

ZebOS-XP ISIS SMI Reference
IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:39 2015

Contents

| | | |
|----------|--|----------|
| 1 | File Index | 1 |
| 1.1 | File List | 1 |
| 2 | File Documentation | 3 |
| 2.1 | smi_isis.h File Reference | 3 |
| 2.1.1 | Detailed Description | 35 |
| 2.1.2 | Function Documentation | 36 |
| 2.1.2.1 | smi_isis_address_family_ipv6_unicast_unset | 36 |
| 2.1.2.2 | smi_isis_adjacency_check_ipv4_set | 36 |
| 2.1.2.3 | smi_isis_adjacency_check_ipv4_unset | 37 |
| 2.1.2.4 | smi_isis_adjacency_check_ipv6_set | 37 |
| 2.1.2.5 | smi_isis_adjacency_check_ipv6_unset | 38 |
| 2.1.2.6 | smi_isis_api_show_ipv6_protocols | 38 |
| 2.1.2.7 | smi_isis_area_password_set | 38 |
| 2.1.2.8 | smi_isis_area_password_unset | 39 |
| 2.1.2.9 | smi_isis_auth_key_chain_set | 39 |
| 2.1.2.10 | smi_isis_auth_key_chain_unset | 40 |
| 2.1.2.11 | smi_isis_auth_mode_hmac_md5_set | 40 |
| 2.1.2.12 | smi_isis_auth_mode_hmac_md5_unset | 41 |
| 2.1.2.13 | smi_isis_auth_mode_text_set | 41 |
| 2.1.2.14 | smi_isis_auth_mode_text_unset | 42 |
| 2.1.2.15 | smi_isis_auth_send_only_set | 42 |
| 2.1.2.16 | smi_isis_auth_send_only_unset | 43 |
| 2.1.2.17 | smi_isis_clear_counters | 43 |
| 2.1.2.18 | smi_isis_clear_interface_counters | 44 |
| 2.1.2.19 | smi_isis_clear_ip_route | 44 |

| | | |
|----------|---|----|
| 2.1.2.20 | smi_isis_clear_ipv6_route | 44 |
| 2.1.2.21 | smi_isis_cspf_set | 45 |
| 2.1.2.22 | smi_isis_cspf_unset | 45 |
| 2.1.2.23 | smi_isis_debug | 46 |
| 2.1.2.24 | smi_isis_default_information_originate_ipv4_set | 46 |
| 2.1.2.25 | smi_isis_default_information_originate_ipv4_unset | 47 |
| 2.1.2.26 | smi_isis_default_information_originate_ipv6_set | 47 |
| 2.1.2.27 | smi_isis_default_information_originate_ipv6_unset | 48 |
| 2.1.2.28 | smi_isis_distance_ipv6_set | 48 |
| 2.1.2.29 | smi_isis_distance_ipv6_unset | 48 |
| 2.1.2.30 | smi_isis_distance_set | 49 |
| 2.1.2.31 | smi_isis_distance_source_set | 49 |
| 2.1.2.32 | smi_isis_distance_source_unset | 50 |
| 2.1.2.33 | smi_isis_distance_unset | 50 |
| 2.1.2.34 | smi_isis_domain_password_set | 51 |
| 2.1.2.35 | smi_isis_domain_password_unset | 51 |
| 2.1.2.36 | smi_isis_get_circ_3way_enabled | 52 |
| 2.1.2.37 | smi_isis_get_circ_adj_changes | 52 |
| 2.1.2.38 | smi_isis_get_circ_admin_state | 53 |
| 2.1.2.39 | smi_isis_get_circ_auth_fails | 53 |
| 2.1.2.40 | smi_isis_get_circ_auth_type_fails | 54 |
| 2.1.2.41 | smi_isis_get_circ_exist_state | 54 |
| 2.1.2.42 | smi_isis_get_circ_ext_domain | 55 |
| 2.1.2.43 | smi_isis_get_circ_id_len_mismatches | 55 |
| 2.1.2.44 | smi_isis_get_circ_ifindex | 56 |
| 2.1.2.45 | smi_isis_get_circ_init_fails | 56 |
| 2.1.2.46 | smi_isis_get_circ_lan_dis_changes | 57 |
| 2.1.2.47 | smi_isis_get_circ_level | 57 |
| 2.1.2.48 | smi_isis_get_circ_level_csnp_interval | 58 |
| 2.1.2.49 | smi_isis_get_circ_level_dis | 58 |
| 2.1.2.50 | smi_isis_get_circ_level_dis_hello_timer | 59 |
| 2.1.2.51 | smi_isis_get_circ_level_hello_multiplier | 59 |
| 2.1.2.52 | smi_isis_get_circ_level_hello_timer | 60 |
| 2.1.2.53 | smi_isis_get_circ_level_id | 60 |

| | | |
|----------|--|----|
| 2.1.2.54 | smi_isis_get_circ_level_id_octet | 61 |
| 2.1.2.55 | smi_isis_get_circ_level_lsp_throttle | 61 |
| 2.1.2.56 | smi_isis_get_circ_level_metric | 62 |
| 2.1.2.57 | smi_isis_get_circ_level_min_lsp_retrans | 62 |
| 2.1.2.58 | smi_isis_get_circ_level_priority | 63 |
| 2.1.2.59 | smi_isis_get_circ_level_psnp_interval | 63 |
| 2.1.2.60 | smi_isis_get_circ_level_wide_metric | 64 |
| 2.1.2.61 | smi_isis_get_circ_max_area_addr_mismatches | 64 |
| 2.1.2.62 | smi_isis_get_circ_mesh_enabled | 65 |
| 2.1.2.63 | smi_isis_get_circ_mesh_group | 65 |
| 2.1.2.64 | smi_isis_get_circ_num_adj | 66 |
| 2.1.2.65 | smi_isis_get_circ_passive_if | 66 |
| 2.1.2.66 | smi_isis_get_circ_rej_adjs | 67 |
| 2.1.2.67 | smi_isis_get_circ_small_hellos | 67 |
| 2.1.2.68 | smi_isis_get_circ_type | 68 |
| 2.1.2.69 | smi_isis_get_circ_uptime | 68 |
| 2.1.2.70 | smi_isis_get_ip_ra_admin_state | 68 |
| 2.1.2.71 | smi_isis_get_ip_ra_exist_state | 69 |
| 2.1.2.72 | smi_isis_get_ip_ra_full_metric | 70 |
| 2.1.2.73 | smi_isis_get_ip_ra_metric | 70 |
| 2.1.2.74 | smi_isis_get_ip_ra_metric_type | 71 |
| 2.1.2.75 | smi_isis_get_ip_ra_snpa_address | 71 |
| 2.1.2.76 | smi_isis_get_ip_ra_source_type | 72 |
| 2.1.2.77 | smi_isis_get_ip_ra_type | 72 |
| 2.1.2.78 | smi_isis_get_is_adj_3way_state | 73 |
| 2.1.2.79 | smi_isis_get_is_adj_area_address | 73 |
| 2.1.2.80 | smi_isis_get_is_adj_extended_circ_id | 74 |
| 2.1.2.81 | smi_isis_get_is_adj_hold_time | 74 |
| 2.1.2.82 | smi_isis_get_is_adj_ip_addr_type | 75 |
| 2.1.2.83 | smi_isis_get_is_adj_ip_address | 76 |
| 2.1.2.84 | smi_isis_get_is_adj_nbr_priority | 76 |
| 2.1.2.85 | smi_isis_get_is_adj_nbr_snpa_addr | 77 |
| 2.1.2.86 | smi_isis_get_is_adj_nbr_sys_id | 77 |
| 2.1.2.87 | smi_isis_get_is_adj_nbr_sys_type | 78 |

| | | |
|-----------|---|----|
| 2.1.2.88 | smi_isis_get_is_adj_prot_supp_protocol | 78 |
| 2.1.2.89 | smi_isis_get_is_adj_state | 79 |
| 2.1.2.90 | smi_isis_get_is_adj_uptime | 79 |
| 2.1.2.91 | smi_isis_get_is_adj_usage | 80 |
| 2.1.2.92 | smi_isis_get_lsp_attributes | 80 |
| 2.1.2.93 | smi_isis_get_lsp_checksum | 81 |
| 2.1.2.94 | smi_isis_get_lsp_lifetime_remain | 81 |
| 2.1.2.95 | smi_isis_get_lsp_pdu_length | 82 |
| 2.1.2.96 | smi_isis_get_lsp_seq | 82 |
| 2.1.2.97 | smi_isis_get_lsp_tlv_checksum | 83 |
| 2.1.2.98 | smi_isis_get_lsp_tlv_index | 83 |
| 2.1.2.99 | smi_isis_get_lsp_tlv_len | 84 |
| 2.1.2.100 | smi_isis_get_lsp_tlv_seq | 84 |
| 2.1.2.101 | smi_isis_get_lsp_tlv_type | 85 |
| 2.1.2.102 | smi_isis_get_lsp_zero_life | 85 |
| 2.1.2.103 | smi_isis_get_man_area_addr_state | 86 |
| 2.1.2.104 | smi_isis_get_packet_count_csnf | 86 |
| 2.1.2.105 | smi_isis_get_packet_count_hello | 87 |
| 2.1.2.106 | smi_isis_get_packet_count_lsp | 87 |
| 2.1.2.107 | smi_isis_get_packet_count_psnf | 88 |
| 2.1.2.108 | smi_isis_get_packet_count_unknown | 88 |
| 2.1.2.109 | smi_isis_get_prot_supp_exist_state | 89 |
| 2.1.2.110 | smi_isis_get_summ_addr_full_metric | 90 |
| 2.1.2.111 | smi_isis_get_summ_addr_metric | 90 |
| 2.1.2.112 | smi_isis_get_summ_addr_state | 91 |
| 2.1.2.113 | smi_isis_get_sys_admin_state | 91 |
| 2.1.2.114 | smi_isis_get_sys_area_addr | 92 |
| 2.1.2.115 | smi_isis_get_sys_exist_state | 92 |
| 2.1.2.116 | smi_isis_get_sys_id | 93 |
| 2.1.2.117 | smi_isis_get_sys_l2_to_l1_leaking | 93 |
| 2.1.2.118 | smi_isis_get_sys_level_lsp_bufsize | 93 |
| 2.1.2.119 | smi_isis_get_sys_level_metric_style | 94 |
| 2.1.2.120 | smi_isis_get_sys_level_min_lsp_gen_interval | 94 |
| 2.1.2.121 | smi_isis_get_sys_level_overload_state | 95 |

| | |
|--|-----|
| 2.1.2.122 smi_isis_get_sys_level_set_overload | 95 |
| 2.1.2.123 smi_isis_get_sys_level_set_overload_until | 96 |
| 2.1.2.124 smi_isis_get_sys_level_spf_considers | 96 |
| 2.1.2.125 smi_isis_get_sys_level_te_enabled | 97 |
| 2.1.2.126 smi_isis_get_sys_log_adj_changes | 97 |
| 2.1.2.127 smi_isis_get_sys_max_age | 98 |
| 2.1.2.128 smi_isis_get_sys_max_area_addrs | 98 |
| 2.1.2.129 smi_isis_get_sys_max_lsp_gen_interval | 98 |
| 2.1.2.130 smi_isis_get_sys_max_path_splits | 99 |
| 2.1.2.131 smi_isis_get_sys_next_circ_index | 99 |
| 2.1.2.132 smi_isis_get_sys_poll_es_hello_rate | 100 |
| 2.1.2.133 smi_isis_get_sys_receive_lsp_bufsize | 100 |
| 2.1.2.134 smi_isis_get_sys_stat_auth_fails | 100 |
| 2.1.2.135 smi_isis_get_sys_stat_auth_type_fails | 101 |
| 2.1.2.136 smi_isis_get_sys_stat_corrupted_lsps | 101 |
| 2.1.2.137 smi_isis_get_sys_stat_exceed_max_seqnums | 102 |
| 2.1.2.138 smi_isis_get_sys_stat_id_len_mismatches | 102 |
| 2.1.2.139 smi_isis_get_sys_stat_lsp_purges | 103 |
| 2.1.2.140 smi_isis_get_sys_stat_lspdb_overloaded | 103 |
| 2.1.2.141 smi_isis_get_sys_stat_man_addr_drop_area | 104 |
| 2.1.2.142 smi_isis_get_sys_stat_max_area_addr_mismatches | 104 |
| 2.1.2.143 smi_isis_get_sys_stat_partition_changes | 105 |
| 2.1.2.144 smi_isis_get_sys_stat_seqnum_skips | 105 |
| 2.1.2.145 smi_isis_get_sys_stat_spf_runs | 106 |
| 2.1.2.146 smi_isis_get_sys_type | 106 |
| 2.1.2.147 smi_isis_get_sys_version | 107 |
| 2.1.2.148 smi_isis_get_sys_wait_time | 107 |
| 2.1.2.149 smi_isis_high_priority_tag_set | 108 |
| 2.1.2.150 smi_isis_high_priority_tag_unset | 108 |
| 2.1.2.151 smi_isis_hostname_dynamic_set | 108 |
| 2.1.2.152 smi_isis_hostname_dynamic_unset | 109 |
| 2.1.2.153 smi_isis_if_auth_key_chain_set | 109 |
| 2.1.2.154 smi_isis_if_auth_key_chain_unset | 110 |
| 2.1.2.155 smi_isis_if_auth_mode_hmac_md5_set | 110 |

| | |
|--|-----|
| 2.1.2.156 smi_isis_if_auth_mode_hmac_md5_unset | 111 |
| 2.1.2.157 smi_isis_if_auth_mode_text_set | 111 |
| 2.1.2.158 smi_isis_if_auth_mode_text_unset | 112 |
| 2.1.2.159 smi_isis_if_auth_send_only_set | 112 |
| 2.1.2.160 smi_isis_if_auth_send_only_unset | 113 |
| 2.1.2.161 smi_isis_if_circuit_type_set | 113 |
| 2.1.2.162 smi_isis_if_circuit_type_unset | 114 |
| 2.1.2.163 smi_isis_if_csnp_interval_set | 114 |
| 2.1.2.164 smi_isis_if_csnp_interval_unset | 115 |
| 2.1.2.165 smi_isis_if_hello_interval_minimal_set | 115 |
| 2.1.2.166 smi_isis_if_hello_interval_set | 116 |
| 2.1.2.167 smi_isis_if_hello_interval_unset | 116 |
| 2.1.2.168 smi_isis_if_hello_multiplier_set | 117 |
| 2.1.2.169 smi_isis_if_hello_multiplier_unset | 117 |
| 2.1.2.170 smi_isis_if_hello_padding_set | 118 |
| 2.1.2.171 smi_isis_if_hello_padding_unset | 118 |
| 2.1.2.172 smi_isis_if_ip_router_set | 118 |
| 2.1.2.173 smi_isis_if_ip_router_unset | 119 |
| 2.1.2.174 smi_isis_if_ipv6_router_set | 119 |
| 2.1.2.175 smi_isis_if_ipv6_router_unset | 120 |
| 2.1.2.176 smi_isis_if_lsp_interval_set | 120 |
| 2.1.2.177 smi_isis_if_lsp_interval_unset | 120 |
| 2.1.2.178 smi_isis_if_mesh_group_block_set | 121 |
| 2.1.2.179 smi_isis_if_mesh_group_set | 121 |
| 2.1.2.180 smi_isis_if_mesh_group_unset | 122 |
| 2.1.2.181 smi_isis_if_metric_set | 122 |
| 2.1.2.182 smi_isis_if_metric_unset | 123 |
| 2.1.2.183 smi_isis_if_network_type_set | 123 |
| 2.1.2.184 smi_isis_if_network_type_unset | 124 |
| 2.1.2.185 smi_isis_if_password_set | 124 |
| 2.1.2.186 smi_isis_if_password_unset | 125 |
| 2.1.2.187 smi_isis_if_priority_set | 125 |
| 2.1.2.188 smi_isis_if_priority_unset | 126 |
| 2.1.2.189 smi_isis_if_retransmit_interval_set | 126 |

| | |
|---|-----|
| 2.1.2.190 smi_isis_if_tag_set | 126 |
| 2.1.2.191 smi_isis_if_tag_unset | 127 |
| 2.1.2.192 smi_isis_if_wide_metric_set | 127 |
| 2.1.2.193 smi_isis_if_wide_metric_unset | 128 |
| 2.1.2.194 smi_isis_ignore_lsp_errors_set | 128 |
| 2.1.2.195 smi_isis_ignore_lsp_errors_unset | 129 |
| 2.1.2.196 smi_isis_instance_set | 129 |
| 2.1.2.197 smi_isis_instance_unset | 130 |
| 2.1.2.198 smi_isis_instance_unset_restart | 130 |
| 2.1.2.199 smi_isis_is_type_set | 130 |
| 2.1.2.200 smi_isis_is_type_unset | 131 |
| 2.1.2.201 smi_isis_ispf_set | 131 |
| 2.1.2.202 smi_isis_ispf_unset | 132 |
| 2.1.2.203 smi_isis_l1_snmp_auth_send_only | 132 |
| 2.1.2.204 smi_isis_l1_snmp_auth_validate_set | 133 |
| 2.1.2.205 smi_isis_l2_snmp_auth_send_only | 133 |
| 2.1.2.206 smi_isis_l2_snmp_auth_validate_set | 134 |
| 2.1.2.207 smi_isis_lsp_gen_interval_set | 134 |
| 2.1.2.208 smi_isis_lsp_gen_interval_unset | 135 |
| 2.1.2.209 smi_isis_lsp_mtu_set | 135 |
| 2.1.2.210 smi_isis_lsp_mtu_unset | 135 |
| 2.1.2.211 smi_isis_lsp_refresh_interval_set | 136 |
| 2.1.2.212 smi_isis_lsp_refresh_interval_unset | 136 |
| 2.1.2.213 smi_isis_max_area_addr_set | 137 |
| 2.1.2.214 smi_isis_max_area_addr_unset | 137 |
| 2.1.2.215 smi_isis_max_lsp_lifetime_set | 138 |
| 2.1.2.216 smi_isis_max_lsp_lifetime_unset | 138 |
| 2.1.2.217 smi_isis_metric_style_set | 138 |
| 2.1.2.218 smi_isis_metric_style_transition_narrow_set | 139 |
| 2.1.2.219 smi_isis_metric_style_transition_set | 139 |
| 2.1.2.220 smi_isis_metric_style_transition_wide_set | 140 |
| 2.1.2.221 smi_isis_metric_style_unset | 141 |
| 2.1.2.222 smi_isis_mpls_traffic_eng_router_id_set | 141 |
| 2.1.2.223 smi_isis_mpls_traffic_eng_router_id_unset | 142 |

| | |
|--|-----|
| 2.1.2.224 smi_isis_mpls_traffic_eng_set | 142 |
| 2.1.2.225 smi_isis_mpls_traffic_eng_unset | 142 |
| 2.1.2.226 smi_isis_multi_topology_set | 143 |
| 2.1.2.227 smi_isis_multi_topology_transition_set | 143 |
| 2.1.2.228 smi_isis_multi_topology_unset | 144 |
| 2.1.2.229 smi_isis_net_set | 144 |
| 2.1.2.230 smi_isis_net_unset | 145 |
| 2.1.2.231 smi_isis_no_debug | 145 |
| 2.1.2.232 smi_isis_parse_sys_id | 146 |
| 2.1.2.233 smi_isis_passive_interface_default_set | 147 |
| 2.1.2.234 smi_isis_passive_interface_default_unset | 147 |
| 2.1.2.235 smi_isis_passive_interface_set | 147 |
| 2.1.2.236 smi_isis_passive_interface_unset | 148 |
| 2.1.2.237 smi_isis_prc_interval_set | 148 |
| 2.1.2.238 smi_isis_proc_clear | 149 |
| 2.1.2.239 smi_isis_protocol_topology_set | 149 |
| 2.1.2.240 smi_isis_protocol_topology_unset | 149 |
| 2.1.2.241 smi_isis_redistribute_inter_level_ipv4_set | 150 |
| 2.1.2.242 smi_isis_redistribute_inter_level_ipv4_unset | 150 |
| 2.1.2.243 smi_isis_redistribute_inter_level_ipv6_set | 151 |
| 2.1.2.244 smi_isis_redistribute_inter_level_ipv6_unset | 151 |
| 2.1.2.245 smi_isis_redistribute_ipv4_set | 152 |
| 2.1.2.246 smi_isis_redistribute_ipv4_unset | 152 |
| 2.1.2.247 smi_isis_redistribute_ipv6_set | 153 |
| 2.1.2.248 smi_isis_redistribute_ipv6_unset | 154 |
| 2.1.2.249 smi_isis_restart_grace_period_set | 154 |
| 2.1.2.250 smi_isis_restart_grace_period_unset | 154 |
| 2.1.2.251 smi_isis_restart_hello_interval_set | 155 |
| 2.1.2.252 smi_isis_restart_hello_interval_unset | 155 |
| 2.1.2.253 smi_isis_restart_helper_set | 156 |
| 2.1.2.254 smi_isis_restart_helper_unset | 156 |
| 2.1.2.255 smi_isis_restart_level_timer_set | 156 |
| 2.1.2.256 smi_isis_restart_level_timer_unset | 157 |
| 2.1.2.257 smi_isis_restart_set | 157 |

| | |
|---|-----|
| 2.1.2.258 smi_isis_restart_suppress_adjacency_set | 158 |
| 2.1.2.259 smi_isis_restart_suppress_adjacency_unset | 158 |
| 2.1.2.260 smi_isis_set_circ_3way_enabled | 158 |
| 2.1.2.261 smi_isis_set_circ_admin_state | 159 |
| 2.1.2.262 smi_isis_set_circ_exist_state | 159 |
| 2.1.2.263 smi_isis_set_circ_ext_domain | 160 |
| 2.1.2.264 smi_isis_set_circ_ifindex | 160 |
| 2.1.2.265 smi_isis_set_circ_level | 160 |
| 2.1.2.266 smi_isis_set_circ_level_dis_hello_timer | 161 |
| 2.1.2.267 smi_isis_set_circ_level_hello_multiplier | 162 |
| 2.1.2.268 smi_isis_set_circ_level_hello_timer | 162 |
| 2.1.2.269 smi_isis_set_circ_level_id_octet | 163 |
| 2.1.2.270 smi_isis_set_circ_level_lsp_throttle | 163 |
| 2.1.2.271 smi_isis_set_circ_level_metric | 164 |
| 2.1.2.272 smi_isis_set_circ_level_wide_metric | 164 |
| 2.1.2.273 smi_isis_set_circ_mesh_enabled | 165 |
| 2.1.2.274 smi_isis_set_circ_mesh_group | 165 |
| 2.1.2.275 smi_isis_set_circ_passive_if | 166 |
| 2.1.2.276 smi_isis_set_circ_small_hellos | 166 |
| 2.1.2.277 smi_isis_set_circ_type | 167 |
| 2.1.2.278 smi_isis_set_ip_ra_admin_state | 167 |
| 2.1.2.279 smi_isis_set_ip_ra_exist_state | 168 |
| 2.1.2.280 smi_isis_set_ip_ra_full_metric | 169 |
| 2.1.2.281 smi_isis_set_ip_ra_metric | 169 |
| 2.1.2.282 smi_isis_set_ip_ra_metric_type | 170 |
| 2.1.2.283 smi_isis_set_ip_ra_nexthop_type | 170 |
| 2.1.2.284 smi_isis_set_ip_ra_type | 171 |
| 2.1.2.285 smi_isis_set_man_area_addr_state | 171 |
| 2.1.2.286 smi_isis_set_prot_supp_exist_state | 172 |
| 2.1.2.287 smi_isis_set_sys_admin_state | 173 |
| 2.1.2.288 smi_isis_set_sys_exist_state | 173 |
| 2.1.2.289 smi_isis_set_sys_l2_to_l1_leaking | 173 |
| 2.1.2.290 smi_isis_set_sys_level_lsp_bufsize | 174 |
| 2.1.2.291 smi_isis_set_sys_level_set_overload | 174 |

| | |
|---|-----|
| 2.1.2.292 smi_isis_set_sys_level_set_overload_until | 175 |
| 2.1.2.293 smi_isis_set_sys_level_spf_considers | 175 |
| 2.1.2.294 smi_isis_set_sys_level_te_enabled | 176 |
| 2.1.2.295 smi_isis_set_sys_log_adj_changes | 176 |
| 2.1.2.296 smi_isis_set_sys_max_age | 177 |
| 2.1.2.297 smi_isis_set_sys_max_area_addrs | 177 |
| 2.1.2.298 smi_isis_set_sys_max_lsp_gen_interval | 178 |
| 2.1.2.299 smi_isis_set_sys_max_path_splits | 178 |
| 2.1.2.300 smi_isis_set_sys_poll_es_hello_rate | 179 |
| 2.1.2.301 smi_isis_set_sys_receive_lsp_bufsize | 179 |
| 2.1.2.302 smi_isis_set_sys_type | 179 |
| 2.1.2.303 smi_isis_set_sys_wait_time | 180 |
| 2.1.2.304 smi_isis_show_clns_if_nbr_api | 180 |
| 2.1.2.305 smi_isis_show_clns_nbr_api | 181 |
| 2.1.2.306 smi_isis_show_clns_neighbors_api | 182 |
| 2.1.2.307 smi_isis_show_database | 182 |
| 2.1.2.308 smi_isis_show_database_filtered | 183 |
| 2.1.2.309 smi_isis_show_global_stat | 183 |
| 2.1.2.310 smi_isis_show_if_stat | 184 |
| 2.1.2.311 smi_isis_show_tag_global_stat | 184 |
| 2.1.2.312 smi_isis_show_tag_if_stat | 185 |
| 2.1.2.313 smi_isis_show_topology_all | 185 |
| 2.1.2.314 smi_isis_spf_interval_set | 186 |
| 2.1.2.315 smi_isis_spf_interval_unset | 186 |
| 2.1.2.316 smi_isis_summary_address_set | 187 |
| 2.1.2.317 smi_isis_summary_address_unset | 187 |
| 2.1.2.318 smi_show_ip_isis_route | 188 |
| 2.1.2.319 smi_show_ipv6_isis_route | 188 |
| 2.1.2.320 smi_show_isis_interface | 188 |

Chapter 1

File Index

1.1 File List

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

[smi_isis.h](#) (Provides API for managing ISIS) 3

Chapter 2

File Documentation

2.1 smi_isis.h File Reference

Provides API for managing ISIS. `#include "smi_client.h"`
`#include "smi_isis_msg.h"`

Functions

- int [smi_show_isis_interface](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, int start_index, int end_index, struct list *ifBriefList, int(*funpointer)(struct list *ifBriefList))
*This function retrieves all interface details and brief details **
- int [smi_show_ip_isis_route](#) (struct smiclient_globals *azg, u_int32_t vr_id, struct list *outputList, u_int32_t(*callbackFunc)(struct list *outputList))
This function displays the isis IP Route Info.
- int [smi_show_ipv6_isis_route](#) (struct smiclient_globals *azg, u_int32_t vr_id, struct list *outputList, u_int32_t(*callbackFunc)(struct list *outputList))
This function displays the isis IPV6 Route Info.
- int [smi_isis_area_password_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *passwd)
This function sets authentication password for an area.
- int [smi_isis_area_password_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)
This function unsets authentication password for an area.
- int [smi_isis_show_clsns_nbr_api](#) (struct smiclient_globals *azg, int vr_id, int start_index, int end_index, char *tag, struct list *isis_clsns_nbr, u_int32_t(*callbackFunc)(struct list *isis_clsns_nbr))

This function retrieves detailed ISIS is-neighbors information.

- int [smi_isis_show_clsns_neighbors_api](#) (struct smiclient_globals *azg, int vr_id, int start_index, int end_index, char *tag, struct list *isis_clsns_nbr, u_int32_t(*callbackFunc)(struct list *isis_clsns_nbr))

This function retrieves detailed ISIS neighbors information.

- int [smi_isis_show_clsns_if_nbr_api](#) (struct smiclient_globals *azg, int vr_id, int start_index, int end_index, char *tag, char *ifname, struct list *isis_clsns_nbr, u_int32_t(*callbackFunc)(struct list *isis_clsns_nbr))

This function retrieves detailed ISIS neighbors information.

- int [smi_isis_auth_mode_text_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function sets the authentication mode to text.

- int [smi_isis_auth_mode_text_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function unset the authentication mode to text.

- int [smi_isis_auth_mode_hmac_md5_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function sets the authentication mode to MD5.

- int [smi_isis_auth_mode_hmac_md5_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function unsets the authentication mode to MD5.

- int [smi_isis_auth_key_chain_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *key_chain, int level)

This function sets the key chain to be used for authentication.

- int [smi_isis_auth_key_chain_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function unsets the key chain to be used for authentication.

- int [smi_isis_auth_send_only_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function configures the send-only option, that is, not to validate the authentication on the received packets.

- int [smi_isis_auth_send_only_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function unconfigure the send-only option, that is, to validate the authentication on the received packets.

- int [smi_isis_l1_snmp_auth_validate_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *passwd)

This function sets the authentication password for the Level-1 SNP(Sequence number PDUs)and check the password in SNPs that it receives.

- int [smi_isis_l1_snp_auth_send_only](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *passwd)

This function sets the authentication password for the Level-1 SNP(Sequence number PDUs) but not check the password in SNP PDUs that it receives.

- int [smi_isis_domain_password_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *passwd)

Set the authentication password for a routing domain.

- int [smi_isis_l2_snp_auth_validate_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *passwd)

This function sets the authentication password for the Level-2 domain and SNP(Sequence number PDUs),also checks the password in SNPs that it receives.

- int [smi_isis_l2_snp_auth_send_only](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *passwd)

This function sets the authentication password for the Level-2 domain and SNP(Sequence number PDUs) but not check the password in SNP PDUs that it receives.

- int [smi_isis_domain_password_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function unsets the authentication password for a routing domain.

- int [smi_isis_ignore_lsp_errors_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function ignores receiving LSPs(Link State Packets) with checksum error. LSP will be accepted as if it is valid.

- int [smi_isis_ignore_lsp_errors_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function validates receiving the LSP checksum. The LSP will be rejected if the checksum has an error.

- int [smi_isis_is_type_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int is_type)

This function sets IS Level routing process as a station router only or as both a station router and an area router or as an area router only.

- int [smi_isis_is_type_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int is_type)

This function reset IS-IS Level to default.

- int [smi_isis_ispf_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int ispf_level)

This function enables incremental SPF for routing process.

- int [smi_isis_ispf_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function unsets incremental SPF for routing process.

- int [smi_isis_prc_interval_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_int32_t min_delay, u_int32_t max_delay)

This function reset parameters for Partial Route Computation (PRC).

- int [smi_isis_lsp_gen_interval_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level, u_char lsp_gen_interval)

This function configures the minimum interval between regenerating the same LSP.

- int [smi_isis_lsp_gen_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function resets the minimum interval between regenerating the same LSP.

- int [smi_isis_max_area_addr_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_char limit)

This function sets the maximum number of ISIS areas that can be configured on a router. By default, ISIS permits a maximum of three areas that can be defined on a router.

- int [smi_isis_max_area_addr_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function sets the maximum number of ISIS areas to its default(3).

- int [smi_isis_lsp_refresh_interval_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_int32_t lsp_refresh_interval)

This function sets the LSP refresh interval.

- int [smi_isis_lsp_refresh_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function resets the LSP refresh interval.

- int [smi_isis_max_lsp_lifetime_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_int32_t max_lifetime)

This function configures the maximum LSP lifetime.

- int [smi_isis_max_lsp_lifetime_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function unconfigure the maximum LSP lifetime, and set it to the default value 1200 (seconds).

- int [smi_isis_net_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *net)

This function configures Network Entity Title (NET) for the process.

- int [smi_isis_net_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *net)

This function unconfigures Network Entity Title (NET) for the process.

- int [smi_isis_spf_interval_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level, u_int32_t start_delay, u_int32_t min_delay, u_int32_t max_delay)

This function configures the minimum and maximum interval between SPF calculations.

- int [smi_isis_spf_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function unconfigure the minimum interval between SPF calculations. Default is 10 (seconds).

- int [smi_isis_lsp_mtu_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int size, int level)

This function sets Link state Packet(lsp) MTU .

- int [smi_isis_lsp_mtu_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function restore Link state Packet(lsp) MTU to default of 1492 bytes.

- int [smi_isis_hostname_dynamic_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int flag)

This function configures the dynamic hostname TLV capability.

- int [smi_isis_hostname_dynamic_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function unconfigures the dynamic hostname TLV capability.

- int [smi_isis_if_auth_mode_text_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function configures the isis authentication mode to text.

- int [smi_isis_if_auth_mode_text_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function unsets the isis authentication mode to text.

- int [smi_isis_if_auth_mode_hmac_md5_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function sets the authentication mode to MD5.

- int [smi_isis_if_auth_mode_hmac_md5_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This call implements the no parameter of the isis authentication mode md5 command to unset the authentication mode to MD5.

- int [smi_isis_if_auth_key_chain_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, char *key_chain, int level)

This function configures the key chain to be used for authentication.

- int [smi_isis_if_auth_key_chain_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function removes the key chain to be used for authentication.

- int [smi_isis_if_auth_send_only_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function configures the send-only option, that is, not to validate the authentication on the hello PDUs.

- int [smi_isis_if_auth_send_only_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function unconfigure the send-only option, that is, to validate the authentication on the hello PDUs.

- int [smi_isis_if_ip_router_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, char *tag)

This function enables IP router interface commands.

- int [smi_isis_if_ip_router_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, char *tag)

This function disable IP router interface commands.

- int [smi_isis_if_ipv6_router_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, char *tag)

This function enables the interface for IPv6 routing.

- int [smi_isis_if_ipv6_router_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, char *tag)

This function disable IPV6 router interface commands.

- int [smi_isis_if_network_type_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int type)

This function sets the IS-IS network type to either point to point or broadcast.

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

This function sets the IS-IS network type to the default value.

- int [smi_isis_if_circuit_type_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int type)

This function sets the interface's circuit type.

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

This function sets the interface's circuit type to default.

- int [smi_isis_if_csnp_interval_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_int32_t if_csnp_interval, int level)

This function sets the complete sequence number PDUs (CSNPs) interval for the interface.

- int [smi_isis_if_csnp_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function unsets the complete sequence number PDUs (CSNPs) interval for the interface.

- int [smi_isis_if_hello_interval_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_int32_t if_hello_interval, int level)

This function configures interface's Hello interval.

- int [smi_isis_if_hello_interval_minimal_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function configures the Holdtime in Hello PDU to 1 second.

- int [smi_isis_if_hello_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function unconfigure interface's Hello interval.

- int [smi_isis_if_hello_multiplier_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_int32_t multi, int level)

This function configures the interface's Hello-Multiplier value.

- int [smi_isis_if_hello_multiplier_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function unconfigure the interface's Hello-Multiplier value. Default value is 3.

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

This function enables IS-IS Hello packet padding.

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

This function disables IS-IS Hello packet padding.

- int [smi_isis_if_lsp_interval_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_int32_t if_lsp_interval)

This function configures the interface's LSP transmission interval.

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

This function unconfigures the interface's LSP transmission interval.

- int [smi_isis_if_metric_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_char metric, int level)

This function configures the interface's metric value.

- int [smi_isis_if_metric_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function unconfigures the interface's metric value.

- int [smi_isis_if_wide_metric_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_int32_t metric, int level)

This function configures the interface's wide metric value.

- int [smi_isis_if_wide_metric_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function unconfigures the interface's wide metric value.

- int [smi_isis_high_priority_tag_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_int32_t priority_tag)

This function sets the high-priority tag.

- int [smi_isis_high_priority_tag_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function unsets the high-priority tag.

- int [smi_isis_if_tag_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_int32_t tag, int level)

This function sets the priority tag.

- int [smi_isis_if_tag_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function unsets the priority tag.

- int [smi_isis_if_password_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, char *passwd, int level)

This function sets the interface's authentication password.

- int [smi_isis_if_password_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function unsets the interface's authentication password.

- int [smi_isis_if_priority_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_char priority, int level)

This function sets the interface's Priority value for Designated Router election.

- int [smi_isis_if_priority_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)

This function unsets the interface's Priority value for Designated Router election.

- int [smi_isis_if_retransmit_interval_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_int32_t if_retransmit_interval)

This function resets the LSP retransmission interval.

- int [smi_isis_if_mesh_group_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_int32_t group_id)

This function configures the mesh group ID.

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

This function configures the interface as mesh-group blocked.

- int [smi_isis_if_mesh_group_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_int32_t group_id)

This function unconfigure the mesh group ID or mesh group blocked.

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

This function creates an IS-IS instance for enabling a routing process.

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

This function deletes an IS-IS instance.

- int [smi_isis_adjacency_check_ipv4_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function enables adjacency check based on the IPv4 protocol TLVs in the IS-IS hello packet.

- int [smi_isis_adjacency_check_ipv4_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function disables adjacency check based on the IPv4 protocol TLVs in the IS-IS Hello packet.

- int [smi_isis_adjacency_check_ipv6_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function enables adjacency check based on the IPv6 protocol TLVs in the IS-IS hello packet.

- int [smi_isis_adjacency_check_ipv6_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function disables adjacency check based on the IPv6 protocol TLVs in the IS-IS Hello packet.

- int [smi_isis_redistribute_ipv4_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int source, u_int32_t metric, u_char metric_type, int level, char *rmap_name)

This function inject IPv4 routes into IS-IS from another routing protocol.

- int [smi_isis_redistribute_ipv4_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int source)

This function stop injecting IPv4 routes into IS-IS from another routing protocol.

- int [smi_isis_redistribute_inter_level_ipv4_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level, char *name)

This function configures inter-level redistribution for IPv4.

- int [smi_isis_redistribute_inter_level_ipv4_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function unconfigures inter-level redistribution for IPv4.

- int [smi_isis_redistribute_ipv6_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int source, u_int32_t metric, u_char metric_type, int level, char *rmap_name)

This function inject IPv6 routes into IS-IS from another routing protocol.

- int [smi_isis_redistribute_ipv6_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int source)

This function stop injecting IPv6 routes into IS-IS from another routing protocol.

- int [smi_isis_redistribute_inter_level_ipv6_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level, char *name)

This function configures inter-level redistribution for IPv6.

- int [smi_isis_redistribute_inter_level_ipv6_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function unconfigures inter-level redistribution for IPv6.

- int [smi_isis_default_information_originate_ipv4_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char origin, char *rmap_name)

This function injects IPv4 default route into IS-IS.

- int [smi_isis_default_information_originate_ipv4_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_char origin, char *rmap_name)

This function stop injecting IPv4 default route into IS-IS.

- int [smi_isis_default_information_originate_ipv6_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char origin, char *rmap_name)

This function injects IPv6 default route into IS-IS.

- int [smi_isis_default_information_originate_ipv6_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_char origin, char *rmap_name)

This function stop injecting IPv6 default route into IS-IS.

- int [smi_isis_metric_style_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function configures the metric style as wide in TLVs.

- int [smi_isis_metric_style_transition_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function configures the metric-style transition in TLVs.

- int [smi_isis_metric_style_transition_narrow_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function configures metric-style as transition narrow in TLVs.

- int [smi_isis_metric_style_transition_wide_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function configures metric-style as transition wide in TLVs.

- int [smi_isis_metric_style_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function unconfigures the metric style in TLVs.

- int [smi_isis_multi_topology_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_char level)

This function configures topology type as multi-topology in TLVs and SPF calculation.

- int [smi_isis_multi_topology_transition_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_char level)

This function configures the topology type as multi-topology transition in TLVs and SPF calculation.

- int [smi_isis_multi_topology_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_char level)

This function configures the topology type as single-topology in TLVs and SPF calculation.

- int [smi_isis_protocol_topology_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_char level)

This function enables Protocol Topology support.

- int [smi_isis_protocol_topology_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_char level)

This function disables Protocol Topology support.

- int [smi_isis_mpls_traffic_eng_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function enables traffic engineering in both level-1 and level-2 routers.

- int [smi_isis_mpls_traffic_eng_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This function disables traffic engineering in both level-1 and level-2 routers.

- int [smi_isis_mpls_traffic_eng_router_id_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, struct pal_in4_addr router_id)

This function configures the TE router-ID.

- int [smi_isis_mpls_traffic_eng_router_id_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function unconfigures the TE router-ID.

- int [smi_isis_cspf_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function creates an IS-IS CSPF server.

- int [smi_isis_cspf_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function deletes an IS-IS CSPF server.

- int [smi_isis_summary_address_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, struct pal_in4_addr addr, u_char masklen, int level, u_int8_t metric)

This call implements the summary-address command to summarize specific IPv4 reachability information.

- int [smi_isis_summary_address_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, struct pal_in4_addr addr, u_char masklen)

This call implements the no parameter of the summary-address command to remove the summary.

- int [smi_isis_show_if_stat](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, struct smi_isis_if_stat *isis_if_stat, u_int32_t(*callbackFunc)(struct smi_isis_if_stat *isis_is_stat))

This call fetchs the isis interface counters of the given interface.

- int [smi_isis_show_tag_if_stat](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, char *tag, struct smi_isis_if_stat *isis_if_stat, u_int32_t(*callbackFunc)(struct smi_isis_if_stat *isis_is_stat))

This call fetchs the isis interface counters of the given interface and tag.

- int [smi_isis_show_global_stat](#) (struct smiclient_globals *azg, int vr_id, int start_index, int end_index, struct list *isisOutList, u_int32_t(*callbackFunc)(struct list *isisOutList))
This call fetchs the isis global counters.
- int [smi_isis_show_tag_global_stat](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, struct smi_isis_global_stat *isis_global_stat, u_int32_t(*callbackFunc)(struct smi_isis_global_stat *isis_global_stat))
This call fetchs the isis global counters of the given tag.
- int [smi_isis_clear_interface_counters](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname)
This call clears isis interface counter.
- int [smi_isis_clear_counters](#) (struct smiclient_globals *azg, u_int32_t vr_id)
This call clears isis counter.
- int [smi_isis_clear_ip_route](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *str)
This call clears ISIS IP local redistribution routes.
- int [smi_isis_proc_clear](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)
This call clears isis process.
- int [smi_isis_address_family_ipv6_unicast_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)
This function summarize specific IPv6 reachability information.
- int [smi_isis_clear_ipv6_route](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *str)
This function clears the IPV6 isis route.
- int [smi_isis_restart_hello_interval_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, u_int16_t restart_hello_interval, int level)
This call implements the isis restart-hello-interval command to configure the interval of the IS-IS Hello packet with Restart TLV.
- int [smi_isis_restart_hello_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *name, int level)
This call implements the no parameter of the isis restart-hello-interval command to reset the interval of the IS-IS Hello packet interval with Restart TLV to the default.
- int [smi_isis_restart_level_timer_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_int16_t timer, int level)
This call implements the restart-timer command to configure the maximum timer to wait for the LSP database synchronization.

- int [smi_isis_restart_level_timer_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int level)

This call implements the no parameter of the restart-timer command to reset the maximum timer to wait for the LSP database synchronization to the default.

- int [smi_isis_restart_grace_period_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t seconds)

This call implements the isis restart grace-period command to configure the grace period.

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

This call implements the no parameter of the isis restart grace-period command to reset to the default value the grace period.

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

This call implements the isis restart helper command to configure the router as the helper router.

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

This call implements the no parameter of the isis restart helper command to unconfigure the router as the helper router. This means that a non-helper router initializes adjacency with the restarting router, and recalculates the topology.

- int [smi_isis_restart_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t seconds)

This call notify NSM to restore the IS-IS routes in the NSM routing table.

- int [smi_isis_instance_unset_restart](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This call implements the part of the restart isis command to force shutdown of the IS-IS instance. This stores routes in the NSM, and shuts down the ISIS daemon.

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

This call restarts suppress-adjacency.

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

This call stops the suppress-adjacency.

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

This call gets the version number of the IS-IS protocol that this instance implements.

- int [smi_isis_get_sys_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *sys_type)
This call gets the system type for the instance of the IS-IS protocol.
- int [smi_isis_get_sys_id](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_char *sys_id)
This call gets the system ID for the instance of the IS-IS protocol.
- int [smi_isis_get_sys_max_path_splits](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *sys_max_path_splits)
This call sets the maximum number of paths with equal routing metric values permitted to split between. Only the default value can be set.
- int [smi_isis_get_sys_max_lsp_gen_interval](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *sys_max_lsp_gen_interval)
This call gets the maximum interval, in seconds, between generated LSPs by this instance of the IS-IS protocol.
- int [smi_isis_get_sys_max_area_addrs](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *sys_max_area_addrs)
This call gets the maximum number of area addresses to be permitted for this instance of the IS-IS protocol.
- int [smi_isis_get_sys_poll_es_hello_rate](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *poll_es_hello_rate)
This call gets the value, in seconds, to be used for the suggested ES configuration timer in ISH PDUs when soliciting the ES configuration.
- int [smi_isis_get_sys_wait_time](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *sys_wait_time)
This call gets the number of seconds to delay in waiting state before entering the on state.
- int [smi_isis_get_sys_admin_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *sys_wait_time)
This call gets the administrative state of this instance of the IS-IS protocol.
- int [smi_isis_get_sys_log_adj_changes](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *sys_log_adj_changes)
This call gets the state of the log generation when an IS-IS adjacency changes state (up or down).
- int [smi_isis_get_sys_next_circ_index](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *sys_next_circ_index)
This call gets the next ISIS circ index value for this instance of the IS-IS protocol.
- int [smi_isis_get_sys_l2_to_l1_leaking](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *sys_l2_to_l1_leaking)

This call gets the state of the level 2 to level 1 route leaking for this instance of the IS-IS protocol.

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

This call gets the system max age value for LSPs generated by this instance of the IS-IS protocol.

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

This call gets the size of the largest buffer this instance can store.

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

This call gets the state of the IS-IS router of this instance.

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

This call sets the system type for the instance of the IS-IS protocol.

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

This call sets the maximum number of paths with equal routing metric values permitted to split between. Only the default value can be set.

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

This call sets the maximum interval, in seconds, between generated LSPs by this instance of the IS-IS protocol. Only the default value can be set.

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

This call sets the maximum number of area addresses to be permitted for this instance of the IS-IS protocol. Only the default value can be set.

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

This call sets the value, in seconds, to be used for the suggested ES configuration timer in ISH PDUs when soliciting the ES configuration. Only the default value can be set.

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

This call sets the seconds to delay in waiting state before entering an on state. Only the default value can be set.

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

This call sets the administrative state of an instance of the IS-IS protocol. Only the default value can be set.

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

This call sets the state of the log generation when an IS-IS adjacency changes state (up or down).

- int [smi_isis_set_sys_l2_to_l1_leaking](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t sys_l2_to_l1_leaking)

This call sets the state of the level 2 to level 1 route leaking, for this instance of the IS-IS protocol.

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

This call sets the value for the RemainingLifeTime field of the LSP, which is generated by an instance of IS-IS.

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

This call sets the size of the largest buffer this instance can store.

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

This call sets the state of the IS-IS router of this instance.

- int [smi_isis_get_man_area_addr_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, struct smi_isis_area_addr area_addr, u_int32_t *man_area_addr_state)

This call gets the state of the manually configured area address.

- int [smi_isis_set_man_area_addr_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, struct smi_isis_area_addr area_addr, u_int32_t man_area_addr_state)

This call sets the state of the manually configured area address.

- int [smi_isis_get_sys_area_addr](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, struct smi_isis_area_addr area_addr, struct smi_isis_area_addr *sys_area_addr)

This call gets the area address reported in a level 1 LSP received by this instance of the IS-IS protocol.

- int [smi_isis_get_prot_supp_exist_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t protocol, u_int32_t *prot_supp_exist_state)

This call gets the state of the supported protocol.

- int [smi_isis_set_prot_supp_exist_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t protocol, u_int32_t prot_supp_exist_state)

This call gets the state of the supported protocol.

- int [smi_isis_get_summ_addr_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t *summ_addr_state)

This call gets the existence state of this summary address.

- int [smi_isis_get_summ_addr_metric](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t *summ_addr_metric)

This call gets the metric value to announce this summary address.

- int [smi_isis_get_summ_addr_full_metric](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t *summ_addr_full_metric)

This call gets the wide metric value to announce this summary address.

- int [smi_isis_get_sys_level_lsp_bufsize](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *sys_level_lsp_bufsize)

This call gets the maximum size of LSPs and SNPs originated by this Intermediate System at this level.

- int [smi_isis_get_sys_level_min_lsp_gen_interval](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *min_lsp_gen_interval)

This call gets Minimum interval, in seconds, between successive generation of LSPs with the same LSPID at this level by this Intermediate System.

- int [smi_isis_get_sys_level_overload_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *level_overload_state)

This function gets the state of the database at this level. The value 'off' indicates that IS-IS is not active at this level. The value 'on' indicates that IS-IS is active at this level and is not overloaded.

- int [smi_isis_get_sys_level_set_overload](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *sys_level_set_overload)

This call gets the state of the overload bit for the instance of the IS-IS protocol at this level.

- int [smi_isis_get_sys_level_set_overload_until](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *sys_level_set_overload_until)

This call sets the time, in seconds, the overload bit should be set for the instance of the IS-IS protocol at this level.

- int [smi_isis_get_sys_level_metric_style](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *sys_level_metric_style)

This call gets the metric style for the instance of the IS-IS protocol at this level.

- int [smi_isis_get_sys_level_spf_considers](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *sys_level_spf_considers)

This call gets the metric to be considered in the SPF computation for the instance of the IS-IS protocol at this level.

- int [smi_isis_get_sys_level_te_enabled](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *sys_level_te_enabled)

This call gets the state of the traffic engineering for the instance of the IS-IS protocol at this level.

- int [smi_isis_set_sys_level_lsp_bufsize](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t sys_level_lsp_bufsize)

This call sets the maximum size of LSPs and SNPs originated by the instance of the IS-IS protocol at this level.

- int [smi_isis_set_sys_level_set_overload](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t sys_level_set_overload)

This call sets the state of the overload bit for the instance of the IS-IS protocol at this level.

- int [smi_isis_set_sys_level_set_overload_until](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t sys_level_set_overload_until)

This call sets the time, in seconds, the overload bit should be set for the instance of the IS-IS protocol at this level.

- int [smi_isis_set_sys_level_spf_considers](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t sys_level_spf_considers)

This call sets the type of metric to consider in the SPF computation for an IS-IS instance at this level.

- int [smi_isis_set_sys_level_te_enabled](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t sys_level_spf_considers)

This call sets the state of the traffic engineering for the instance of the IS-IS protocol at this level.

- int [smi_isis_get_circ_ifindex](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_ifindex)

This call gets the value of interface index for the interface to which this circuit corresponds.

- `int smi_isis_get_circ_admin_state` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_admin_state)

This call gets the administrative state of the circuit.

- `int smi_isis_get_circ_exist_state` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_exist_state)

This call gets the existence state of the circuit.

- `int smi_isis_get_circ_type` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_type)

This call gets the type of a circuit.

- `int smi_isis_get_circ_ext_domain` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_ext_domain)

This call gets the status of the normal transmission and interpretation of intra-domain IS-IS PDUs on this circuit.

- `int smi_isis_get_circ_level` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_level)

This call gets the type of packets that will be sent and accepted on this circuit.

- `int smi_isis_get_circ_passive_if` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_passive_if)

This call gets the status to include this circuit in LSPs, even if it is not running the IS-IS protocol.

- `int smi_isis_get_circ_mesh_enabled` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_mesh_enabled)

This call gets the status of the mesh group configuration of this circuit.

- `int smi_isis_get_circ_mesh_group` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_mesh_group)

This call gets the identifier of the mesh group of this circuit.

- `int smi_isis_get_circ_small_hellos` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_small_hellos)

This call gets the status of the IS-IS LAN hellos padding of this circuit.

- `int smi_isis_get_circ_uptime` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_uptime)

This call gets the amount of time, in seconds, since this circuit entered state 'up' if the circuit is up, or the number of seconds since the circuit was up if the circuit is not up, or since the system started if the circuit has never been up.

- `int smi_isis_get_circ_3way_enabled` (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *circ_3way_enabled)

This call gets the status of this circuit enabled 3Way handshake.

- int [smi_isis_set_circ_ifindex](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_ifindex)

This call sets the value of interface index for an interface for a corresponding circuit. The interface index cannot be changed.

- int [smi_isis_set_circ_admin_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_admin_state)

This call sets the administrative state of the circuit.

- int [smi_isis_set_circ_exist_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_exist_state)

This call sets the existence state of the circuit.

- int [smi_isis_set_circ_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_type)

This call sets the type of the circuit. only broadcast and point-to-point type circuits are supported.

- int [smi_isis_set_circ_ext_domain](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_ext_domain)

This call sets the status of the normal transmission of and interpretation of intra-domain IS-IS PDUs on this circuit.

- int [smi_isis_set_circ_level](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_level)

This call sets the type of packets that will be sent and accepted on this circuit.

- int [smi_isis_set_circ_passive_if](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_passive_if)

This call sets the status to include this circuit in LSPs, even if it is not running the IS-IS protocol.

- int [smi_isis_set_circ_mesh_enabled](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_mesh_enabled)

This call sets the status of the mesh group configuration of this circuit.

- int [smi_isis_set_circ_mesh_group](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_mesh_group)

This call sets the identifier of the mesh group of this circuit.

- int [smi_isis_set_circ_small_hellos](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_small_hellos)

This call sets the status of the IS-IS LAN hellos padding of this circuit.

- int [smi_isis_set_circ_3way_enabled](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_3way_enabled)

This call sets the status of this circuit enabled 3Way handshake.

- int [smi_isis_get_circ_level_metric](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_metric)

This call gets the metric value of this circuit for this level.

- int [smi_isis_get_circ_level_wide_metric](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_wide_metric)

This call gets the wide metric value of this circuit for this level.

- int [smi_isis_get_circ_level_priority](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_priority)

This call gets the priority for becoming the LAN designated IS at this level.

- int [smi_isis_get_circ_level_id_octet](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_id_octet)

This call gets a one-byte identifier that can be used in protocol packets to identify a circuit for this level.

- int [smi_isis_get_circ_level_id](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, struct smi_isis_dis_id *circ_level_id)

This call gets the ID of the circuit allocated during initialization.

- int [smi_isis_get_circ_level_dis](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, struct smi_isis_dis_id *circ_level_dis)

This call gets the ID of the LAN designated IS on this circuit at this level.

- int [smi_isis_get_circ_level_hello_multiplier](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_hello_multiplier)

This call gets the hello multiplier that is multiplied by the corresponding HelloTimer; and the result in seconds (rounded up) is used as the holding time in transmitted hellos, to be used by receivers of hello packets from this IS.

- int [smi_isis_get_circ_level_hello_timer](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_hello_timer)

This call gets the maximum period, in milliseconds, between IIH PDUs on multi-access networks at this level for LANs. The value at level 1 is used as the period between Hellos on L1L2 point-to-point circuits.

- int [smi_isis_get_circ_level_dis_hello_timer](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_dis_hello_timer)
This call gets the period, in milliseconds, between hello PDUs on multiaccess networks when this is the designated IS.
- int [smi_isis_get_circ_level_lsp_throttle](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_lsp_throttle)
This call gets the minimal interval of time, in milliseconds, between transmissions of LSPs on an interface at this level.
- int [smi_isis_get_circ_level_min_lsp_retrans](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_min_lsp_retrans)
This call gets the minimum interval, in seconds, between re-transmission of an LSP at this level.
- int [smi_isis_get_circ_level_csnp_interval](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_csnp_interval)
This call gets the interval of time, in seconds, between transmission of CSNPs on multiaccess networks if this router is the designated IS at this level.
- int [smi_isis_get_circ_level_psnp_interval](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t *circ_level_psnp_interval)
This call gets the minimum interval in seconds between sending PSNP at this level. PSNP interval switch is not supported.
- int [smi_isis_set_circ_level_metric](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_metric)
This call sets the metric value of this circuit for this level.
- int [smi_isis_set_circ_level_wide_metric](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_wide_metric)
This call sets the wide metric value of this circuit for this level.
- int [smi_isis_set_circ_level_id_octet](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_id_octet)
This call sets a one-byte identifier that is used in protocol packets to identify a circuit for this level. The level ID octet cannot be changed.
- int [smi_isis_set_circ_level_hello_multiplier](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_hello_multiplier)

This call sets the hello multiplier which is multiplied by the corresponding HelloTimer, and the result, in seconds (rounded up), is used as the holding time in transmitted hellos, to be used by receivers of hello packets from this IS.

- int [smi_isis_set_circ_level_hello_timer](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_hello_timer)

This call sets the maximum period, in milliseconds, between IIH PDUs on multiaccess networks at this level for LANs. The value at level 1 is used as the period between Hellos on L1L2 point-to-point circuits.

- int [smi_isis_set_circ_level_dis_hello_timer](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_dis_hello_timer)

This call sets the period, in milliseconds, between hello PDUs on multiaccess networks when this is the designated IS.

- int [smi_isis_set_circ_level_lsp_throttle](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_lsp_throttle)

This call sets minimal interval of time, in milliseconds, between transmissions of LSPs on an interface at this level.

- int [smi_isis_get_sys_stat_corrupted_lsps](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *stat_corrupted_lsps)

This call gets the number of corrupted in-memory LSPs detected.

- int [smi_isis_get_sys_stat_auth_type_fails](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *stat_auth_type_fails)

This call gets the number of authentication type mismatches.

- int [smi_isis_get_sys_stat_auth_fails](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *stat_auth_fails)

This call gets the number of authentication failures.

- int [smi_isis_get_sys_stat_lspdb_overloaded](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *sys_stat_lspdb_overloaded)

This call gets the number of times the LSP database has become overloaded.

- int [smi_isis_get_sys_stat_man_addr_drop_area](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *stat_man_addr_drop_area)

This call gets the number of times a manual address has been dropped from the area.

- int [smi_isis_get_sys_stat_exceed_max_seqnums](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *stat_exceed_max_seqnums)

This call gets the number of times the IS has attempted to exceed the maximum sequence number.

- int [smi_isis_get_sys_stat_seqnum_skips](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *stat_seqnum_skips)

This call gets the number of times a sequence number skip has occurred.

- int [smi_isis_get_sys_stat_lsp_purges](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *sys_stat_lsp_purges)

This call gets the number of times a zero-aged copy of the system's own LSP is received from another node.

- int [smi_isis_get_sys_stat_id_len_mismatches](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *stat_id_len_mismatches)

This call gets the number of times a PDU is received with a different value for ID field length to that of the receiving system.

- int [smi_isis_get_sys_stat_max_area_addr_mismatches](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *stat_max_area_addr_mismatches)

This call gets the number of times a PDU is received with a different value for MaximumAreaAddresses from that of the receiving system.

- int [smi_isis_get_sys_stat_partition_changes](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *stat_partition_changes)

This call gets the number of times partition changes occurred.

- int [smi_isis_get_sys_stat_spf_runs](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *stat_spf_runs)

This call gets the number of times SPF ran at this level.

- int [smi_isis_get_circ_adj_changes](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circetype, u_int32_t *circ_adj_changes)

This call gets the number of times an adjacency stat change has occurred on this circuit.

- int [smi_isis_get_circ_num_adj](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circetype, u_int32_t *circ_num_adj)

This call gets the number of adjacencies on this circuit.

- int [smi_isis_get_circ_init_fails](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circetype, u_int32_t *circ_init_fails)

This call gets the number of times initialization of this circuit has failed.

- int [smi_isis_get_circ_rej_adjs](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circetype, u_int32_t *circ_rej_adjs)

This call gets the number of times an adjacency has been rejected on this circuit.

- int [smi_isis_get_circ_id_len_mismatches](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circetype, u_int32_t *circ_id_len_mismatches)

This call gets the number of times an IS-IS control PDU with an ID field length different from that of this system has been received.

- int [smi_isis_get_circ_max_area_addr_mismatches](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circetype, u_int32_t *circ_max_area_addr_mismatches)

This call gets the number of times an IS-IS control PDU with a max area address field different from that of this system has been received.

- int [smi_isis_get_circ_auth_type_fails](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circetype, u_int32_t *circ_auth_type_fails)

This call gets the number of times an IS-IS control PDU with an auth type field different from that of this system has been received.

- int [smi_isis_get_circ_auth_fails](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circetype, u_int32_t *circ_auth_fails)

This call gets the number of times an IS-IS control PDU with the correct auth type has failed to pass authentication validation.

- int [smi_isis_get_circ_lan_dis_changes](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circetype, u_int32_t *circ_lan_dis_changes)

This call gets the number of times an adjacency stat change has occurred on the next circuit level.

- int [smi_isis_get_packet_count_hello](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t direction, u_int32_t *packet_count_hello)

This call gets the number of IS-IS Hello PDUs seen in this direction at this level.

- int [smi_isis_get_packet_count_lsp](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t direction, u_int32_t *packet_count_lsp)

This call gets the number of IS-IS LSPs seen in this direction at this level.

- int [smi_isis_get_packet_count_csnp](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t direction, u_int32_t *packet_count_csnp)

This call gets the number of IS-IS CSNPs seen in this direction at this level.

- int [smi_isis_get_packet_count_psnp](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t direction, u_int32_t *packet_count_psnp)

This call gets the number of IS-IS PSNPs seen in this direction at this level.

- int [smi_isis_get_packet_count_unknown](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t direction, u_int32_t *packet_count_unknown)

This call gets the number of unknown IS-IS PDUs seen in this direction at this level.

- int [smi_isis_get_is_adj_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t *is_adj_state)

This call gets the state of the adjacency.

- int [smi_isis_get_is_adj_3way_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t *is_adj_3way_state)

This call gets the 3Way state of the adjacency.

- int [smi_isis_get_is_adj_nbr_snpa_addr](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_char *is_adj_nbr_snpa_addr)

This call gets the SNPA address of the neighboring IS.

- int [smi_isis_get_is_adj_nbr_sys_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t *is_adj_nbr_sys_type)

This call gets the type of the neighboring IS.

- int [smi_isis_get_is_adj_extended_circ_id](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t *is_adj_extended_circ_id)

This call gets the four byte extended circuit ID learned from the Neighbor during 3-way handshake, or 0.

- int [smi_isis_get_is_adj_nbr_sys_id](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_char *is_adj_nbr_sys_id)

This call gets the system ID of the neighboring IS.

- int [smi_isis_get_is_adj_usage](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t *is_adj_usage)

This call gets the adjacency usage with the neighboring IS.

- int [smi_isis_get_is_adj_hold_time](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t *is_adj_hold_time)

This call gets the holding time in seconds for this adjacency.

- int [smi_isis_get_is_adj_nbr_priority](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t *is_adj_nbr_priority)

This call gets the priority of the neighboring IS for becoming the designated IS.

- int [smi_isis_get_is_adj_uptime](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t *is_adj_uptime)

This call gets the amount of time in seconds since this adjacency entered 'up'.

- int [smi_isis_get_is_adj_area_address](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circuit_id, u_int32_t adjindex, u_int32_t areaindex, struct smi_isis_area_addr *is_adj_area_address)

This call gets one area address as reported in IIH PDUs received from the adjacent neighbor.

- int [smi_isis_get_is_adj_ip_addr_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t ipindex, u_int32_t *is_adj_ip_addr_type)

This call gets the type of one IP address as reported in IIH PDUs received from the adjacent neighbor.

- int [smi_isis_get_is_adj_ip_address](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t ipindex, struct prefix *is_adj_ip_address)

This call gets one IP address as reported in IIH PDUs received from the adjacent neighbor.

- int [smi_isis_get_is_adj_prot_supp_protocol](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t protocol, u_int32_t *is_adj_prot_supp_protocol)

This call gets the type of network protocol supported by the adjacent neighbor.

- int [smi_isis_get_ip_ra_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t *ip_ra_type)

This call gets the type of this IP Reachable Address.

- int [smi_isis_get_ip_ra_exist_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t *ip_ra_exist_state)

This call gets the state of this IP Reachable Address.

- int [smi_isis_get_ip_ra_admin_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t *ip_ra_admin_state)

This call gets the administrative state of the IP Reachable Address.

- int [smi_isis_get_ip_ra_metric](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t *ip_ra_metric)

This call gets the metric value for reaching the specified destination over this circuit.

- int [smi_isis_get_ip_ra_metric_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t *ip_ra_metric_type)

This call gets the type of metric which indicates whether the metric is internal or external.

- int [smi_isis_get_ip_ra_full_metric](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t *ip_ra_full_metric)

This call gets the wide metric value for reaching the specified destination over this circuit.

- int [smi_isis_get_ip_ra_snpa_address](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_char *ip_ra_snpa_address)

This call gets the SNPA Address to which a PDU may be forwarded in order to reach a destination that matches this IP Reachable Address.

- int [smi_isis_get_ip_ra_source_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t *ip_ra_source_type)

This call gets the origin of this route.

- int [smi_isis_set_ip_ra_nexthop_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_nexthop_type)

This call sets the type of the IP nexthop address.

- int [smi_isis_set_ip_ra_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_type)

This call sets the type of this IP Reachable Address.

- int [smi_isis_set_ip_ra_exist_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_exist_state)

This call sets the state of this IP Reachable Address.

- int [smi_isis_set_ip_ra_admin_state](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_admin_state)

This call set the administrative state of the IP Reachable Address.

- int [smi_isis_set_ip_ra_metric](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_metric)

This call sets the metric value for reaching the specified destination over this circuit.

- int [smi_isis_set_ip_ra_metric_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_metric_type)

This call sets the type of metric that indicates whether the metric is internal or external.

- int [smi_isis_set_ip_ra_full_metric](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_full_metric)

This call sets the wide metric value for reaching the specified destination over this circuit.

- int [smi_isis_get_lsp_seq](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t *lsp_seq_num)

This call gets the sequence number for this LSP.

- int [smi_isis_get_lsp_zero_life](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t *lsp_state)

This call gets the state that indicates whether or not this LSP is being purged by this system.

- int [smi_isis_get_lsp_checksum](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t *lsp_checksum)

This call gets the 16-bit fletcher checksum for this LSP.

- int [smi_isis_get_lsp_lifetime_remain](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t *lsp_lifetime)

This call gets the remaining lifetime, in seconds, for this LSP.

- int [smi_isis_get_lsp_pdu_length](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t *lsp_len)

This call gets the length of this LSP.

- int [smi_isis_get_lsp_attributes](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t *lsp_flags)

This call gets the flags carried by this LSP.

- int [smi_isis_get_lsp_tlv_index](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t tlvinde, u_int32_t *lsp_index)

This call gets the index of this TLV in the LSP. This object follows the index behavior.

- int [smi_isis_get_lsp_tlv_seq](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t tlvinde, u_int32_t *lsp_seq)

This call gets the sequence number for this LSP.

- int [smi_isis_get_lsp_tlv_checksum](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t tlvinde, u_int32_t *lsp_fletcher_checksum)

This call gets the 16-bit Fletcher checksum for this LSP.

- int [smi_isis_get_lsp_tlv_type](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t tlvinde, u_int32_t *tlvtype)

This call gets the type of this TLV.

- int [smi_isis_get_lsp_tlv_len](#) (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t tlvinde, u_int32_t *tlv_len)

This call gets the length of this TLV.

- int [smi_isis_distance_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_int32_t distance)

This call implements the distance command to define an administrative distance for all routes from a specific source and/or all routes permitted by an access-list.

- int [smi_isis_distance_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This call implements the distance command to define an administrative distance for all routes from a specific source and/or all routes permitted by an access-list.

- int [smi_isis_distance_source_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_int32_t distance, char *sys_id, char *access_name)

This call implements the distance command to define an administrative distance for all routes from a specific route source and/or all routes permitted by an access-list.

- int [smi_isis_distance_source_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *sys_id, char *access_name)

This call implements the no distance command to remove an administrative distance for all a specific routes from a specific source and/or all routes permitted by an access-list.

- int [smi_isis_distance_ipv6_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, u_int32_t distance)

This call implements the distance command, which defines an administrative distance for all routes for an IPv6 address family.

- int [smi_isis_distance_ipv6_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This call implements the no distance command to remove an administrative distance for all routes for an IPv6 address family.

- int [smi_isis_passive_interface_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *name)

This call sets the interface to passive mode for the current interface.

- int [smi_isis_passive_interface_default_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This call sets all interfaces into passive mode, except the highpriority interface.

- int [smi_isis_passive_interface_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, char *name)

This call resets the interface to active mode for the current interface.

- int [smi_isis_passive_interface_default_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This call resets all interfaces to active mode.

- int [smi_isis_parse_sys_id](#) (struct smiclient_globals *azg, char *arg, u_char *sys_id)

This function parses system ID.

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

Use this function to turn on debugging for specified criteria. It enable to show some debugging information about specified criteria into file.

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

Use this function to turn off debugging for specified criteria.

- int [smi_isis_show_counter](#) (struct smiclient_globals *azg, int vr_id, int start_index, int end_index, struct list *isOutList, u_int32_t(*callbackFunc)(struct list *isOutList))

- int **smi_isis_show_interface_counter** (struct smiclient_globals *azg, int vr_id, char *ifname, int start_index, int end_index, struct list *isisOutList, u_int32_t (*callbackFunc)(struct list *isisOutList))
- int **smi_isis_get_clsns_is_neighbors_ifname** (struct smiclient_globals *azg, u_int32_t vr_id, u_char *ifname, u_char *tag, int start_index, int end_index, struct list *smi_isis_clsns_is_nbr_if, int (*funpointer)(struct list *smi_isis_clsns_is_nbr_if))
- int **smi_isis_show_database** (struct smiclient_globals *azg, int vr_id, int start_index, int end_index, char *flag, struct list *isisOutList, u_int32_t (*callbackFunc)(struct list *isisOutList))

This call displays the entire ISIS database.

- int **smi_isis_show_database_filtered** (struct smiclient_globals *azg, int vr_id, char *tag, char *lspid, char *level, char *flag, struct list *isisOutList, u_int32_t (*callbackFunc)(struct list *isisOutList))

This call displays the filtered ISIS database.

- int **smi_isis_api_show_ip_protocols** (struct smiclient_globals *azg, u_int32_t vr_id, int start_index, int end_index, struct list *isisOutList, u_int32_t (*callbackFunc)(struct list *isisOutList))
- int **smi_isis_api_show_ipv6_protocols** (struct smiclient_globals *azg, u_int32_t vr_id, int start_index, int end_index, struct list *isisOutList, u_int32_t (*callbackFunc)(struct list *isisOutList))

This call displays ISIS protocol related information.

- int **smi_isis_show_topology_all** (struct smiclient_globals *azg, u_int32_t vr_id, int pindex, char *tag, int level, struct list *isisOutlist, u_int32_t (*callbackFunc)(struct list *isisOutlist))

This call displays ISIS ipv6 protocol related information.

- int **smi_isis_show_tag_topology** (struct smiclient_globals *azg, u_int32_t vr_id, int pindex, char *tag, int level, struct list *isisOutlist, u_int32_t (*callbackFunc)(struct list *isisOutlist))

2.1.1 Detailed Description

Provides API for managing ISIS. The API provided in this file forms the basis of ZebOS ISIS management. These APIs are used by various north bound management interfaces like CLI, SNMP and SMI The Intermediate System-to-Intermediate System (IS-IS) protocol is a two-level hierarchical interior gateway protocol (IGP) for routing both IP and OSI, using a link-state in the individual areas that make up the hierarchy. The Shortest Path First (SPF) computation is used to calculate the shortest path tree (SPT) inside each area.

2.1.2 Function Documentation

2.1.2.1 `int smi_isis_address_family_ipv6_unicast_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function summarize specific IPv6 reachability information. `smi_isis_summary_prefix_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *addr* IPv6 network address
- ← *masklen* Mask length
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2
- ← *metric* Metric value

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE

`smi_isis_address_family_ipv6_unicast_unset`

This function configures the address family mode.

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.2 `int smi_isis_adjacency_check_ipv4_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function enables adjacency check based on the IPv4 protocol TLVs in the IS-IS hello packet. `smi_isis_adjacency_check_ipv4_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

**2.1.2.3 int smi_isis_adjacency_check_ipv4_unset (struct smiclient_globals *
azg, u_int32_t vr_id, char * tag)**

This function disables adjacency check based on the IPv4 protocol TLVs in the IS-IS Hello packet. smi_isis_adjacency_check_ipv4_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

**2.1.2.4 int smi_isis_adjacency_check_ipv6_set (struct smiclient_globals * azg,
u_int32_t vr_id, char * tag)**

This function enables adjacency check based on the IPv6 protocol TLVs in the IS-IS hello packet. smi_isis_adjacency_check_ipv6_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.5 `int smi_isis_adjacency_check_ipv6_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function disables adjacency check based on the IPv6 protocol TLVs in the IS-IS Hello packet. `smi_isis_adjacency_check_ipv6_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.6 `int smi_isis_api_show_ipv6_protocols (struct smiclient_globals * azg, u_int32_t vr_id, int start_index, int end_index, struct list * isOutList, u_int32_t(*) (struct list * isOutList) callbackFunc)`

This call displays ISIS protocol related information. `smi_isis_api_show_ip_protocols`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *start_index*
- ← *end_index* index
- *isOutList* Pointer to linked list of structure `isis_infolist`
- *callbackFunc* Callback function

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.7 `int smi_isis_area_password_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char * passwd)`

This function sets authentication password for an area. `smi_isis_area_password_set`

Parameters:

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

- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *passwd* Authentication key, null-terminated

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_PASSWORD_TOO_LONG
 ISIS_API_SET_ERR_AUTH_MD5_EXIST
 ISIS_API_SET_ERR_AUTH_TEXT_EXIST

2.1.2.8 int smi_isis_area_password_unset (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*)

This function unsets authentication password for an area. smi_isis_area_password_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.9 int smi_isis_auth_key_chain_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, char * *key_chain*, int *level*)

This function sets the key chain to be used for authentication. smi_isis_auth_key_chain_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *key_chain* Key chain used for authentication
- ← *level* IS-IS instance level (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_NO_AUTH_MD5_OR_TEXT_EXIST

**2.1.2.10 int smi_isis_auth_key_chain_unset (struct smiclient_globals * azg,
 u_int32_t vr_id, char * tag, int level)**

This function unsets the key chain to be used for authentication. smi_isis_auth_key_chain_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

**2.1.2.11 int smi_isis_auth_mode_hmac_md5_set (struct smiclient_globals * azg,
 u_int32_t vr_id, char * tag, int level)**

This function sets the authentication mode to MD5. smi_isis_auth_mode_hmac_md5_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_AUTH_TEXT_EXIST

2.1.2.12 int smi_isis_auth_mode_hmac_md5_unset (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, int *level*)

This function unsets the authentication mode to MD5. smi_isis_auth_mode_hmac_md5_unset

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual Router Id
 ← *tag* IS-IS instance area tag
 ← *level* IS-IS instance level (1|2|3)
 1 Level-1
 2 Level-2
 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.13 int smi_isis_auth_mode_text_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, int *level*)

This function sets the authentication mode to text. smi_isis_auth_mode_text_set

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual Router Id
 ← *tag* IS-IS instance area tag
 ← *level* IS-IS instance level (1|2|3)
 1 Level-1
 2 Level-2
 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_AUTH_MD5_EXIST

**2.1.2.14 int smi_isis_auth_mode_text_unset (struct smiclient_globals * azg,
 u_int32_t vr_id, char * tag, int level)**

This function unset the authentication mode to text. smi_isis_auth_mode_text_unset

Parameters:

← **azg** Pointer to the SMI client global structure
 ← **vr_id** Virtual Router Id
 ← **tag** IS-IS instance area tag
 ← **level** IS-IS instance level (1|2|3)
 1 Level-1
 2 Level-2
 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

**2.1.2.15 int smi_isis_auth_send_only_set (struct smiclient_globals * azg,
 u_int32_t vr_id, char * tag, int level)**

This function configures the send-only option, that is, not to validate the authentication on the received packets. smi_isis_auth_send_only_set

Parameters:

← **azg** Pointer to the SMI client global structure
 ← **vr_id** Virtual Router Id
 ← **tag** IS-IS instance area tag
 ← **level** IS-IS instance level (1|2|3)
 1 Level-1
 2 Level-2
 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

**2.1.2.16 int smi_isis_auth_send_only_unset (struct smiclient_globals * azg,
 u_int32_t vr_id, char * tag, int level)**

This function unconfigure the send-only option, that is, to validate the authentication on the received packets. smi_isis_auth_send_only_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

**2.1.2.17 int smi_isis_clear_counters (struct smiclient_globals * azg, u_int32_t
 vr_id)**

This call clears isis counter. smi_isis_clear_counters

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for vr_id

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.18 `int smi_isis_clear_interface_counters (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname)`

This call clears isis interface counter. `smi_isis_clear_interface_counters`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for *vr_id*
- ← *ifname* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_IF_NOT_EXIST

2.1.2.19 `int smi_isis_clear_ip_route (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char * str)`

This call clears ISIS IP local redistribution routes. `smi_isis_clear_ip_route`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for *vr_id*
- ← *tag* IS-IS instance area tag
- ← *str* Input string

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.20 `int smi_isis_clear_ipv6_route (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char * str)`

This function clears the IPV6 isis route. `smi_isis_clear_ipv6_route`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

← *str*

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
ISIS_API_SET_ERR_INVALID_VALUE
ISIS_API_SET_ERROR

2.1.2.21 int smi_isis_cspf_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)

This function creates an IS-IS CSPF server. smi_isis_cspf_set

Parameters:

← *azg* Pointer to the SMI client global structure
← *vr_id* Virtual Router Id
← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
ISIS_API_SET_ERR_TE_NOT_ENABLED
ISIS_API_SET_ERR_CSPF_INSTANCE_EXIST
ISIS_API_SET_ERR_CSPF_INSTANCE_EXIST

2.1.2.22 int smi_isis_cspf_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)

This function deletes an IS-IS CSPF server. smi_isis_cspf_unset

Parameters:

← *azg* Pointer to the SMI client global structure
← *vr_id* Virtual Router Id
← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
ISIS_API_SET_ERR_CSPF_DISABLE_FAILED

2.1.2.23 `int smi_isis_debug (struct smiclient_globals * azg, int vr_id, int debug)`

Use this function to turn on debugging for specified criteria. It enable to show some debugging information about specified criteria into file. `smi_isis_debug`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual router id
- ← **debug** Pass debug flag as following:
 - SMI_ISIS_DEBUG_ALL - Enables all debugging
 - SMI_ISIS_DEBUG_IFSM - Debugging for interface finite state machine
 - SMI_ISIS_DEBUG_NFSM - Debugging for neighbor finite state machine
 - SMI_ISIS_DEBUG_PDU - Debugging for protocol data unit
 - SMI_ISIS_DEBUG_LSP - Debugging for label switched path
 - SMI_ISIS_DEBUG_SPF - Debugging for shortest path first route calculation
 - SMI_ISIS_DEBUG_CHECKSUM - Debugging for checksums
 - SMI_ISIS_DEBUG_AUTH - Debugging for authentication
 - SMI_ISIS_DEBUG_LOCUPD - Debugging for local updates
 - SMI_ISIS_DEBUG_PROTOERROR - Debugging for protocol errors
 - SMI_ISIS_DEBUG_HELLO - Debugging for hello processing
 - SMI_ISIS_DEBUG_EVENTS - Debugging for internal events
 - SMI_ISIS_DEBUG_NSM - Debugging for NSM messages
 - SMI_ISIS_DEBUG_RIB - Debugging for RIB messages
 - SMI_ISIS_DEBUG_BFD - Debugging for bidirectional forwarding detection
 - SMI_ISIS_DEBUG_MPLS - Debugging for multiprotocol label switching

Returns:

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

2.1.2.24 `int smi_isis_default_information_originate_ipv4_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char origin, char * rmap_name)`

This function injects IPv4 default route into IS-IS. `smi_isis_default_information_originate_ipv4_set`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router Id
- ← **tag** IS-IS instance area tag
- ← **origin** origin

← *rmap_name* Name of route-map

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.25 `int smi_isis_default_information_originate_ipv4_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, u_char origin, char * rmap_name)`

This function stop injecting IPv4 default route into IS-IS. smi_isis_default_information_originate_ipv4_unset

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual Router Id
 ← *tag* IS-IS instance area tag
 ← *origin* origin
 ← *rmap_name* Name of route-map

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.26 `int smi_isis_default_information_originate_ipv6_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char origin, char * rmap_name)`

This function injects IPv6 default route into IS-IS. smi_isis_default_information_originate_ipv6_set

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual Router Id
 ← *tag* IS-IS instance area tag
 ← *origin* origin
 ← *rmap_name* Name of route-map

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.27 `int smi_isis_default_information_originate_ipv6_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, u_char origin, char * rmap_name)`

This function stop injecting IPv6 default route into IS-IS. `smi_isis_default_information_originate_ipv6_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *origin* origin
- ← *rmap_name* Name of route-map

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.28 `int smi_isis_distance_ipv6_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, u_int32_t distance)`

This call implements the distance command, which defines an administrative distance for all routes for an IPv6 address family. `smi_isis_distance_ipv6_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *tag* IS-IS instance area tag
- ← *distance* Administrative distance

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_DISTANCE_INVALID
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.29 `int smi_isis_distance_ipv6_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This call implements the no distance command to remove an administrative distance for all routes for an IPv6 address family. `smi_isis_distance_ipv6_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_DISTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.30 int smi_isis_distance_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, u_int32_t *distance*)

This call implements the distance command to define an administrative distance for all routes from a specific source and/or all routes permitted by an access-list. smi_isis_distance_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *tag* IS-IS instance area tag
- ← *distance* Administrative distance

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_DISTANCE_INVALID
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.31 int smi_isis_distance_source_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, u_int32_t *distance*, char * *sys_id*, char * *access_name*)

This call implements the distance command to define an administrative distance for all routes from a specific route source and/or all routes permitted by an access-list. smi_isis_distance_source_set

Parameters:

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

- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *tag* IS-IS instance area tag
- ← *distance* Administrative distance
- ← *sys_id* Source ID
- ← *access_name* Access-list name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_DISTANCE_INVALID
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.32 int smi_isis_distance_source_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char * sys_id, char * access_name)

This call implements the no distance command to remove an administrative distance for all a specific routes from a specific source and/or all routes permitted by an access-list.
 smi_isis_distance_source_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *tag* IS-IS instance area tag
- ← *sys_id* Source ID
- ← *access_name* Access-list name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_DISTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.33 int smi_isis_distance_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)

This call implements the distance command to define an administrative distance for all routes from a specific source and/or all routes permitted by an access-list. smi_isis_distance_unset

Parameters:

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

- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_DISTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.34 int smi_isis_domain_password_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, char * *passwd*)

Set the authentication password for a routing domain. smi_isis_domain_password_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *passwd* Authentication key, null-terminated

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_PASSWORD_TOO_LONG
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_AUTH_TEXT_EXIST
 ISIS_API_SET_ERR_AUTH_MD5_EXIST

2.1.2.35 int smi_isis_domain_password_unset (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*)

This function unsets the authentication password for a routing domain. smi_isis_domain_password_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.36 `int smi_isis_get_circ_3way_enabled (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t * circ_3way_enabled)`

This call gets the status of this circuit enabled 3Way handshake. `smi_isis_get_circ_3way_enabled`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_3way_enabled* Status of the circuit enabled 3Way handshake.
 - 1 isisTruthValueTrue
 - 2 isisTruthValueFalse (default)

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.37 `int smi_isis_get_circ_adj_changes (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_type, u_int32_t * circ_adj_changes)`

This call gets the number of times an adjacency stat change has occurred on this circuit. `smi_isis_get_circ_adj_changes`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *circ_type* Integer that contains the IS-IS circuit type
- *circ_adj_changes* Integer that contains the number of times adjacency state change

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.38 `int smi_isis_get_circ_admin_state (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *
circ_admin_state)`

This call gets the administrative state of the circuit. `smi_isis_get_circ_admin_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_admin_state* Administrative state, including:
 - 1 isisAdminStateOn (default)
 - 2 isisAdminStateOff

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.39 `int smi_isis_get_circ_auth_fails (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
cirttype, u_int32_t * circ_auth_fails)`

This call gets the number of times an IS-IS control PDU with the correct auth type has failed to pass authentication validation. `smi_isis_get_circ_auth_fails`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *cirttype* Integer that contains the IS-IS circuit type
- *circ_auth_fails* Integer that contains the number of times an IS-IS control PDU with the correct auth type has failed to pass authentication validation

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.40 `int smi_isis_get_circ_auth_type_fails (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circtype, u_int32_t * circ_auth_type_fails)`

This call gets the number of times an IS-IS control PDU with an auth type field different from that of this system has been received. `smi_isis_get_circ_auth_type_fails`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *circtype* Integer that contains the IS-IS circuit type
- *circ_auth_type_fails* Integer that contains the number of times an IS-IS control PDU with an auth type field different from that for this system has been received

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.41 `int smi_isis_get_circ_exist_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t * circ_exist_state)`

This call gets the existence state of the circuit. `smi_isis_get_circ_exist_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_exist_state* State of the specified circuit, including:
 - 1 isisRowStatusActive (default)
 - 2 isisRowStatusNotInservice
 - 3 isisRowStatusNotReady
 - 4 isisRowStatusCreateAndGo
 - 5 isisRowStatusCreateAndWait
 - 6 isisRowStatusDestroy

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.42 `int smi_isis_get_circ_ext_domain (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *
circ_ext_domain)`

This call gets the status of the normal transmission and interpretation of intra-domain IS-IS PDUs on this circuit. `smi_isis_get_circ_ext_domain`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_ext_domain* State of the intra-domain IS-IS PDUs, including:
 - 1 isisTruthValueFalse (default)
 - 2 isisTruthValueTrue

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.43 `int smi_isis_get_circ_id_len_mismatches (struct smiclient_globals
* azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex,
u_int32_t circirtype, u_int32_t * circ_id_len_mismatches)`

This call gets the number of times an IS-IS control PDU with an ID field length different from that of this system has been received. `smi_isis_get_circ_id_len_mismatches`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *cirtype* Integer that contains the IS-IS circuit type
- *circ_id_len_mismatches* Integer that contains the number of times an IS-IS control PDU with an ID field length different from that for this system has been received

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.44 `int smi_isis_get_circ_ifindex (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t * circ_ifindex)`

This call gets the value of interface index for the interface to which this circuit corresponds. `smi_isis_get_circ_ifindex`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_ifindex* Interface index that corresponds to the circuit index

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.45 `int smi_isis_get_circ_init_fails (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circtype, u_int32_t * circ_init_fails)`

This call gets the number of times initialization of this circuit has failed. `smi_isis_get_circ_init_fails`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *circtype* Integer that contains the IS-IS circuit type
- *circ_init_fails* Integer that contains the number of times initialization of this circuit has failed

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.46 `int smi_isis_get_circ_lan_dis_changes (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
circ_type, u_int32_t * circ_lan_dis_changes)`

This call gets the number of times an adjacency stat change has occurred on the next circuit level. `smi_isis_get_circ_lan_dis_changes`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *circ_type* Integer that contains the IS-IS circuit type
- *circ_lan_dis_changes* Integer that contains the number of times the designated IS has changed on this circuit at this level. If the circuit is point to point, this count is zero

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.47 `int smi_isis_get_circ_level (struct smiclient_globals * azg, u_int32_t
vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t * circ_level)`

This call gets the type of packets that will be sent and accepted on this circuit. `smi_isis_get_circ_level`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_level* Level of the circuit, including:
 - 1 Level1
 - 2 Level2
 - 3 Level1 and Level 2

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.48 `int smi_isis_get_circ_level_csnp_interval (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t * circ_level_csnp_interval)`

This call gets the interval of time, in seconds, between transmission of CSNPs on multiaccess networks if this router is the designated IS at this level. `smi_isis_get_circ_level_csnp_interval`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_csnp_interval* Integer that contains the CSNP interval.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.49 `int smi_isis_get_circ_level_dis (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, struct smi_isis_dis_id * circ_level_dis)`

This call gets the ID of the LAN designated IS on this circuit at this level. `smi_isis_get_circ_level_dis`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_dis* Pointer to the LAN designated IS ID.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.50 `int smi_isis_get_circ_level_dis_hello_timer (struct smiclient_globals
* azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex,
u_int32_t level, u_int32_t * circ_level_dis_hello_timer)`

This call gets the period, in milliseconds, between hello PDUs on multiaccess networks when this is the designated IS. `smi_isis_get_circ_level_dis_hello_timer`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_dis_hello_timer* Integer that contains the hello timer of designated IS

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.51 `int smi_isis_get_circ_level_hello_multiplier (struct smiclient_globals
* azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex,
u_int32_t level, u_int32_t * circ_level_hello_multiplier)`

This call gets the hello multiplier that is multiplied by the corresponding HelloTimer; and the result in seconds (rounded up) is used as the holding time in transmitted hellos, to be used by receivers of hello packets from this IS. `smi_isis_get_circ_level_hello_multiplier`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_hello_multiplier* Integer that contains the hello multiplier

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.52 `int smi_isis_get_circ_level_hello_timer (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t * circ_level_hello_timer)`

This call gets the maximum period, in milliseconds, between IIH PDUs on multi-access networks at this level for LANs. The value at level 1 is used as the period between Hellos on L1L2 point-to-point circuits. `smi_isis_get_circ_level_hello_timer`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← **instance** Integer that contains the IS-IS instance ID
- ← **circindex** An integer that contains the IS-IS circuit index
- ← **level** An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- **circ_level_hello_timer** Period, in milliseconds, between IIH PDUs on multi-access networks at this level for LANs. The value at level 1 is used as the period between Hellos on L1L2 point to point circuits.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.53 `int smi_isis_get_circ_level_id (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, struct smi_isis_dis_id * circ_level_id)`

This call gets the ID of the circuit allocated during initialization. `smi_isis_get_circ_level_id`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← **instance** Integer that contains the IS-IS instance ID
- ← **circindex** An integer that contains the IS-IS circuit index
- ← **level** An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- **circ_level_id** Pointer to the ID for a circuit.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.54 `int smi_isis_get_circ_level_id_octet (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t * circ_level_id_octet)`

This call gets a one-byte identifier that can be used in protocol packets to identify a circuit for this level. `smi_isis_get_circ_level_id_octet`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 ,For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_id_octet* Integer containing a 1-byte identifier. It can be used in protocol packets to identify a circuit

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.55 `int smi_isis_get_circ_level_lsp_throttle (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t * circ_level_lsp_throttle)`

This call gets the minimal interval of time, in milliseconds, between transmissions of LSPs on an interface at this level. `smi_isis_get_circ_level_lsp_throttle`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_lsp_throttle* Integer that contains LSP minimum interval

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.56 `int smi_isis_get_circ_level_metric (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
level, u_int32_t * circ_level_metric)`

This call gets the metric value of this circuit for this level. `smi_isis_get_circ_level_metric`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_metric* Integer sub-range for default metric for single hop which picks between 0 to 63

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.57 `int smi_isis_get_circ_level_min_lsp_retrans (struct smiclient_globals
* azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex,
u_int32_t level, u_int32_t * circ_level_min_lsp_retrans)`

This call gets the minimum interval, in seconds, between re-transmission of an LSP at this level. `smi_isis_get_circ_level_min_lsp_retrans`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_min_lsp_retrans* Integer that contains the minimum LSP re-transmission interval

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.58 `int smi_isis_get_circ_level_priority (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t * circ_level_priority)`

This call gets the priority for becoming the LAN designated IS at this level. `smi_isis_get_circ_level_priority`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_priority* Integer sub-range for IS-IS priority.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.59 `int smi_isis_get_circ_level_psnp_interval (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t * circ_level_psnp_interval)`

This call gets the minimum interval in seconds between sending PSNP at this level. PSNP interval switch is not supported. `smi_isis_get_circ_level_psnp_interval`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_psnp_interval* Integer that contains the PSNP interval. 2 is returned by default

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.60 `int smi_isis_get_circ_level_wide_metric (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t * circ_level_wide_metric)`

This call gets the wide metric value of this circuit for this level. `smi_isis_get_circ_level_wide_metric`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *circ_level_wide_metric* Wide metric for IS neighbors which pick between 0 to 1,677,215

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.61 `int smi_isis_get_circ_max_area_addr_mismatches (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circitype, u_int32_t * circ_max_area_addr_mismatches)`

This call gets the number of times an IS-IS control PDU with a max area address field different from that of this system has been received. `smi_isis_get_circ_max_area_addr_mismatches`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *circitype* Integer that contains the IS-IS circuit type
- *circ_max_area_addr_mismatches* Integer that contains the number of times an IS-IS control PDU with a max area address field different from that for this system has been received

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.62 `int smi_isis_get_circ_mesh_enabled (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t * circ_mesh_enabled)`

This call gets the status of the mesh group configuration of this circuit. `smi_isis_get_circ_mesh_enabled`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_mesh_enabled* To include this circuit in LSPs, even if it is not running the IS-IS protocol, including:
 - 1 isisMeshGroupInactive
 - 2 isisMeshGroupBlocked
 - 3 isisMeshGroupSet

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.63 `int smi_isis_get_circ_mesh_group (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t * circ_mesh_group)`

This call gets the identifier of the mesh group of this circuit. `smi_isis_get_circ_mesh_group`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_mesh_group* Integer value that represents mesh group ID

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.64 `int smi_isis_get_circ_num_adj (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
circtype, u_int32_t * circ_num_adj)`

This call gets the number of adjacencies on this circuit. `smi_isis_get_circ_num_adj`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *circtype* Integer that contains the IS-IS circuit type
- *circ_num_adj* Integer that contains the number of adjacencies

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.65 `int smi_isis_get_circ_passive_if (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *
circ_passive_if)`

This call gets the status to include this circuit in LSPs, even if it is not running the IS-IS protocol. `smi_isis_get_circ_passive_if`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_passive_if* To include this circuit in LSPs, even if it is not running the IS-IS protocol, including:
 - 1 isisTruthValueFalse (default)
 - 2 isisTruthValueTrue

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.66 `int smi_isis_get_circ_rej_adjs (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
circtype, u_int32_t * circ_rej_adjs)`

This call gets the number of times an adjacency has been rejected on this circuit. `smi_isis_get_circ_rej_adjs`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *circtype* Integer that contains the IS-IS circuit type
- *circ_rej_adjs* Integer that contains the number of times an adjacency has been rejected on this circuit

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.67 `int smi_isis_get_circ_small_hellos (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t *
circ_small_hellos)`

This call gets the status of the IS-IS LAN hellos padding of this circuit. `smi_isis_get_circ_small_hellos`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_small_hellos* Value indicates whether unpadded hellos can be sent on LAN circuits

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.68 `int smi_isis_get_circ_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t * circ_type)`

This call gets the type of a circuit. `smi_isis_get_circ_type`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_type* Type of the specified circuit

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.69 `int smi_isis_get_circ_uptime (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t * circ_uptime)`

This call gets the amount of time, in seconds, since this circuit entered state 'up' if the circuit is up, or the number of seconds since the circuit was up if the circuit is not up, or since the system started if the circuit has never been up. `smi_isis_get_circ_uptime`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- *circ_uptime* Seconds since the object has been 'up'. If the object is not up, seconds since the circuit was up or since the system started if the circuit has never been up

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.70 `int smi_isis_get_ip_ra_admin_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t * ip_ra_admin_state)`

This call gets the administrative state of the IP Reachable Address. `smi_isis_get_ip_ra_admin_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- *ip_ra_admin_state* State of this IP reachable address including : isisRowStatusActive , isisRowStatusNotInservice, isisRowStatusNotReady isisRowStatusCreateAndGo, isisRowStatusCreateAndWait ,isisRowStatusDestroy

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.71 `int smi_isis_get_ip_ra_exist_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t * ip_ra_exist_state)`

This call gets the state of this IP Reachable Address. smi_isis_get_ip_ra_exist_state

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- *ip_ra_exist_state* State of this IP reachable address including : isisRowStatusActive , isisRowStatusNotInservice, isisRowStatusNotReady isisRowStatusCreateAndGo, isisRowStatusCreateAndWait ,isisRowStatusDestroy

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.72 `int smi_isis_get_ip_ra_full_metric (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t * ip_ra_full_metric)`

This call gets the wide metric value for reaching the specified destination over this circuit. `smi_isis_get_ip_ra_full_metric`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← **instance** Integer that contains the IS-IS instance ID
- ← **raindex** Identifier to specify isisIPRAEntry
- ← **type** Type of IP Reachable Address
- ← **summ_ip_addr** Destination of IP Reachable Address
- ← **prefixlen** Length of the IP netmask of IP Reachable Address
- **ip_ra_full_metric** Wide metric value for reaching the specified destination over this circuit

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.73 `int smi_isis_get_ip_ra_metric (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t * ip_ra_metric)`

This call gets the metric value for reaching the specified destination over this circuit. `smi_isis_get_ip_ra_metric`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← **instance** Integer that contains the IS-IS instance ID
- ← **raindex** Identifier to specify isisIPRAEntry
- ← **type** Type of IP Reachable Address
- ← **summ_ip_addr** Destination of IP Reachable Address
- ← **prefixlen** Length of the IP netmask of IP Reachable Address
- **ip_ra_metric** Metric value for reaching the specified destination over this circuit

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.74 `int smi_isis_get_ip_ra_metric_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t * ip_ra_metric_type)`

This call gets the type of metric which indicates whether the metric is internal or external. smi_isis_get_ip_ra_metric_type

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- *ip_ra_metric_type* Type of metric, including: internal and external

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.75 `int smi_isis_get_ip_ra_snpa_address (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_char * ip_ra_snpa_address)`

This call gets the SNPA Address to which a PDU may be forwarded in order to reach a destination that matches this IP Reachable Address. smi_isis_get_ip_ra_snpa_address

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry

- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- *ip_ra_snpa_address* Pointer to the SNPA address to which a PDU may be forwarded to reach a destination

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.76 `int smi_isis_get_ip_ra_source_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t * ip_ra_source_type)`

This call gets the origin of this route. `smi_isis_get_ip_ra_source_type`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- *ip_ra_source_type* Origin of this route

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.77 `int smi_isis_get_ip_ra_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t * ip_ra_type)`

This call gets the type of this IP Reachable Address. `smi_isis_get_ip_ra_type`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*

- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- *ip_ra_type* Type of this IP reachable address. Those of type manual are created by the network manager. Those of type automatic are created through propagation of routing information from another routing protocol.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.78 `int smi_isis_get_is_adj_3way_state (struct smiclient_globals * azg,
 u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
 adjindex, u_int32_t * is_adj_3way_state)`

This call gets the 3Way state of the adjacency. smi_isis_get_is_adj_3way_state

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- *is_adj_3way_state* Integer that contains the 3way state of the adjacency

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.79 `int smi_isis_get_is_adj_area_address (struct smiclient_globals *
 azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circuit_id,
 u_int32_t adjindex, u_int32_t areaindex, struct smi_isis_area_addr *
 is_adj_area_address)`

This call gets one area address as reported in IIH PDUs received from the adjacent neighbor. smi_isis_get_is_adj_area_address

Parameters:

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

- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circuit_id* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- ← *areaindex* Integer that contains the area index associated with area address advertised by the adjacent neighbor
- *is_adj_area_address* Pointer to one area address as reported in IIH PDUs received from the adjacent neighbor

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.80 `int smi_isis_get_is_adj_extended_circ_id (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t * is_adj_extended_circ_id)`

This call gets the four byte extended circuit ID learned from the Neighbor during 3-way handshake, or 0. smi_isis_get_is_adj_extended_circ_id

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- *is_adj_extended_circ_id* Integer that contains the 4-byte extended circuit ID learned from the neighbor during 3- way handshake or 0. Output is always 0

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.81 `int smi_isis_get_is_adj_hold_time (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t * is_adj_hold_time)`

This call gets the holding time in seconds for this adjacency. smi_isis_get_is_adj_hold_time

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- *is_adj_hold_time* Integer that contains the holding time in seconds for this adjacency. This value is based on received IIH PDUs and the elapsed time since receipt.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.82 `int smi_isis_get_is_adj_ip_addr_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t ipindex, u_int32_t * is_adj_ip_addr_type)`

This call gets the type of one IP address as reported in IIH PDUs received from the adjacent neighbor. smi_isis_get_is_adj_ip_addr_type

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- ← *ipindex* Integer that contains the IP address index associated with IP address advertised by the adjacent neighbor.
- *is_adj_ip_addr_type* Integer that contains the type of one IP address as reported in IIH PDUs received from the adjacent neighbor

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.83 `int smi_isis_get_is_adj_ip_address (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t ipindex, struct prefix * is_adj_ip_address)`

This call gets one IP address as reported in IIH PDUs received from the adjacent neighbor. `smi_isis_get_is_adj_ip_address`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- ← *ipindex* Integer that contains the IP address index associated with IP address advertised by the adjacent neighbor
- *is_adj_ip_address* Pointer to prefix structure that contains IP address as reported in IIH PDUs received from the adjacent neighbor

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.84 `int smi_isis_get_is_adj_nbr_priority (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t * is_adj_nbr_priority)`

This call gets the priority of the neighboring IS for becoming the designated IS. `smi_isis_get_is_adj_nbr_priority`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- *is_adj_nbr_priority* Integer that contains priority of the neighboring IS for becoming the designated IS.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.85 `int smi_isis_get_is_adj_nbr_snpa_addr (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
adjindex, u_char * is_adj_nbr_snpa_addr)`

This call gets the SNPA address of the neighboring IS. `smi_isis_get_is_adj_nbr_snpa_addr`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- *is_adj_nbr_snpa_addr* Pointer to the binary SNPA address of the neighboring IS

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.86 `int smi_isis_get_is_adj_nbr_sys_id (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
adjindex, u_char * is_adj_nbr_sys_id)`

This call gets the system ID of the neighboring IS. `smi_isis_get_is_adj_nbr_sys_id`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- *is_adj_nbr_sys_id* Pointer to the system ID of the neighboring IS

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.87 `int smi_isis_get_is_adj_nbr_sys_type (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
adjindex, u_int32_t * is_adj_nbr_sys_type)`

This call gets the type of the neighboring IS. `smi_isis_get_is_adj_nbr_sys_type`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- *is_adj_nbr_sys_type* Integer that contains the type of the neighboring IS, including: Level 1 intermediate system, Level 2 intermediate system, Level 1 and L2 intermediate system on a point-to-point circuit.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.88 `int smi_isis_get_is_adj_prot_supp_protocol (struct smiclient_globals
* azg, u_int32_t vr_id, u_int32_t instance, u_int32_t
circindex, u_int32_t adjindex, u_int32_t protocol, u_int32_t *
is_adj_prot_supp_protocol)`

This call gets the type of network protocol supported by the adjacent neighbor. `smi_isis_get_is_adj_prot_supp_protocol`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- ← *protocol* Integer that contains the supported protocol advertised by the adjacent neighbor
- *is_adj_prot_supp_protocol* Integer that contains the supported protocol advertised by the adjacent neighbor

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.89 `int smi_isis_get_is_adj_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t * is_adj_state)`

This call gets the state of the adjacency. `smi_isis_get_is_adj_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- *is_adj_state* Integer that contains the state of the adjacency

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.90 `int smi_isis_get_is_adj_uptime (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t * is_adj_uptime)`

This call gets the amount of time in seconds since this adjacency entered 'up'. `smi_isis_get_is_adj_uptime`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- *is_adj_uptime* Integer that contains the amount of time in seconds since this adjacency entered state 'up' if the adjacency is up. If the adjacency is not up, the number of seconds since the adjacency was up, or zero, if the adjacency has never been up since the system started.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.91 `int smi_isis_get_is_adj_usage (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t adjindex, u_int32_t * is_adj_usage)`

This call gets the adjacency usage with the neighboring IS. `smi_isis_get_is_adj_usage`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *adjindex* Integer that contains the IS-IS adjacent index
- *is_adj_usage* Integer that contains the adjacency usage with the neighboring IS. Level1 is used for level 1 traffic only. An adjacency of type level2 is used for level 2 traffic only. An adjacency of type level1and2 is used for both level 1 and level 2 traffic on a point-to-point link. There may be two adjacencies (of types level1 and level2) between the same pair of ISs. Level1, Level2,Level1and2

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.92 `int smi_isis_get_lsp_attributes (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t * lsp_flags)`

This call gets the flags carried by this LSP. `smi_isis_get_lsp_attributes`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- *lsp_flags* Flags carried by this LSP

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.93 `int smi_isis_get_lsp_checksum (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct
smi_isis_lspid lspid, u_int32_t * lsp_checksum)`

This call gets the 16-bit fletcher checksum for this LSP. `smi_isis_get_lsp_checksum`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- *lsp_checksum* Checksum for this LSP

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.94 `int smi_isis_get_lsp_lifetime_remain (struct smiclient_globals *
azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct
smi_isis_lspid lspid, u_int32_t * lsp_lifetime)`

This call gets the remaining lifetime, in seconds, for this LSP. `smi_isis_get_lsp_lifetime_remain`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- *lsp_lifetime* Remaining lifetime in seconds for this LSP.

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
ISIS_API_GET_ERROR

2.1.2.95 `int smi_isis_get_lsp_pdu_length (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t * lsp_len)`

This call gets the length of this LSP. `smi_isis_get_lsp_pdu_length`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- *lsp_len* Length of this LSP

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.96 `int smi_isis_get_lsp_seq (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t * lsp_seq_num)`

This call gets the sequence number for this LSP. `smi_isis_get_lsp_seq`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- *lsp_seq_num* Sequence number for this LSP

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.97 `int smi_isis_get_lsp_tlv_checksum (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t tlvindex, u_int32_t * lsp_fletcher_checksum)`

This call gets the 16-bit Fletcher checksum for this LSP. `smi_isis_get_lsp_tlv_checksum`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- ← *tlvindex* Index of this TLV
- *lsp_fletcher_checksum* 16-bit Fletcher checksum for this LSP

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.98 `int smi_isis_get_lsp_tlv_index (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t tlvindex, u_int32_t * lsp_index)`

This call gets the index of this TLV in the LSP. This object follows the index behavior. `smi_isis_get_lsp_tlv_index`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- ← *tlvindex* Index of this TLV
- *lsp_index* Index of this TLV in the LSP

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.99 `int smi_isis_get_lsp_tlv_len (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t tlindex, u_int32_t * tlv_len)`

This call gets the length of this TLV. `smi_isis_get_lsp_tlv_len`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- ← *tlindex* Index of this TLV
- *tlv_len* Length of this TLV in this LSP

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.100 `int smi_isis_get_lsp_tlv_seq (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t tlindex, u_int32_t * lsp_seq)`

This call gets the sequence number for this LSP. `smi_isis_get_lsp_tlv_seq`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- ← *tlindex* Index of this TLV
- *lsp_seq* Sequence number for this LSP

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.101 `int smi_isis_get_lsp_tlv_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t tlvindex, u_int32_t * tlvtype)`

This call gets the type of this TLV. smi_isis_get_lsp_tlv_type

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- ← *tlvindex* Index of this TLV
- *tlvtype* The type of this TLV in this LSP

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.102 `int smi_isis_get_lsp_zero_life (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, struct smi_isis_lspid lspid, u_int32_t * lsp_state)`

This call gets the state that indicates whether or not this LSP is being purged by this system. smi_isis_get_lsp_zero_life

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* IS-IS level this LSP belongs to
- ← *lspid* LSP ID for this LSP
- *lsp_state* State indicating whether or not this LSP is being purged by this system, including: 1 Purged , 2 Not purged

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.103 `int smi_isis_get_man_area_addr_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, struct smi_isis_area_addr area_addr, u_int32_t * man_area_addr_state)`

This call gets the state of the manually configured area address. `smi_isis_get_man_area_addr_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *area_addr* A variable length of a manually configured area address
- *man_area_addr_state* State of the manually configured area address, including:
 - 1 Active (default)
 - 2 Not in service
 - 3 Not ready
 - 4 Create and go
 - 5 Create and wait
 - 6 Destroy

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.104 `int smi_isis_get_packet_count_csnp (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t direction, u_int32_t * packet_count_csnp)`

This call gets the number of IS-IS CSNPs seen in this direction at this level. `smi_isis_get_packet_count_csnp`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *level* Integer that contains the IS-IS level index, including: Level1, Level2
- ← *direction* Integer that contains the packet direction, including: Sending, Receiving
- *packet_count_csnp* Integer that contains the number of IS-IS CSNPs

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.105 `int smi_isis_get_packet_count_hello (struct smiclient_globals * azg,
 u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
 level, u_int32_t direction, u_int32_t * packet_count_hello)`

This call gets the number of IS-IS Hello PDUs seen in this direction at this level. `smi_isis_get_packet_count_hello`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *level* Integer that contains the IS-IS level index, including: Level1, Level2
- ← *direction* Integer that contains the packet direction, including: Sending, Receiving
- *packet_count_hello* Integer that contains the number of IS-IS Hello PDUs

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.106 `int smi_isis_get_packet_count_lsp (struct smiclient_globals * azg,
 u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
 level, u_int32_t direction, u_int32_t * packet_count_lsp)`

This call gets the number of IS-IS LSPs seen in this direction at this level. `smi_isis_get_packet_count_lsp`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *level* Integer that contains the IS-IS level index, including: Level1, Level2
- ← *direction* Integer that contains the packet direction, including: Sending, Receiving

→ *packet_count_lsp* Integer that contains the number of IS-IS LSPs

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.107 `int smi_isis_get_packet_count_psnp (struct smiclient_globals * azg,
 u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
 level, u_int32_t direction, u_int32_t * packet_count_psnp)`

This call gets the number of IS-IS PSNPs seen in this direction at this level. `smi_isis_get_packet_count_psnp`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* Integer that contains the IS-IS circuit index
- ← *level* Integer that contains the IS-IS level index, including: Level1, Level2
- ← *direction* Integer that contains the packet direction, including: Sending, Receiving
- *packet_count_psnp* Integer that contains the number of IS-IS PSNPs

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.108 `int smi_isis_get_packet_count_unknown (struct smiclient_globals
 * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t
 circindex, u_int32_t level, u_int32_t direction, u_int32_t *
 packet_count_unknown)`

This call gets the number of unknown IS-IS PDUs seen in this direction at this level. `smi_isis_get_packet_count_unknown`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID

- ← *circindex* Integer that contains the IS-IS circuit index
- ← *level* Integer that contains the IS-IS level index, including:Level1, Level2
- ← *direction* Integer that contains the packet direction, including:Sending, Receiving
- *packet_count_unknown* Integer that contains the number of unknown IS-IS PDUs

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.109 `int smi_isis_get_prot_supp_exist_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t protocol, u_int32_t * prot_supp_exist_state)`

This call gets the state of the supported protocol. smi_isis_get_prot_supp_exist_state

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *protocol* An integer that contains the supported protocol, including the following values:
 - 129 ISO8473
 - 204 IP
 - 142 IPv6
- *prot_supp_exist_state* State of the manually configured supported protocol, including:
 - 1 Active (default)
 - 2 Not in service
 - 3 Not ready
 - 4 Create and go
 - 5 Create and wait
 - 6 Destroy

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.110 `int smi_isis_get_summ_addr_full_metric (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t * summ_addr_full_metric)`

This call gets the wide metric value to announce this summary address. `smi_isis_get_summ_addr_full_metric`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *type* Type of summary IP address
- ← *summ_ip_addr* Summary IP address
- ← *prefixlen* Prefix length of summary IP address
- *summ_addr_full_metric* Wide metric value to announce this summary address

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.111 `int smi_isis_get_summ_addr_metric (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t * summ_addr_metric)`

This call gets the metric value to announce this summary address. `smi_isis_get_summ_addr_metric`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *type* Type of summary IP address
- ← *summ_ip_addr* Summary IP address
- ← *prefixlen* Prefix length of summary IP address
- *summ_addr_metric* Metric value to announce this summary address

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.112 `int smi_isis_get_summ_addr_state (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t type, struct prefix
summ_ip_addr, u_int32_t prefixlen, u_int32_t * summ_addr_state)`

This call gets the existence state of this summary address. `smi_isis_get_summ_addr_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *type* Type of summary IP address
- ← *summ_ip_addr* Summary IP address
- ← *prefixlen* Prefix length of summary IP address
- *summ_addr_state* Existence state of this summary address. Active is returned by default

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.113 `int smi_isis_get_sys_admin_state (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_wait_time)`

This call gets the administrative state of this instance of the IS-IS protocol. `smi_isis_get_sys_admin_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_wait_time* Administrative state
 - 1 On (default)
 - 2 Off

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.114 `int smi_isis_get_sys_area_addr (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, struct smi_isis_area_addr
area_addr, struct smi_isis_area_addr * sys_area_addr)`

This call gets the area address reported in a level 1 LSP received by this instance of the IS-IS protocol. `smi_isis_get_sys_area_addr`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *area_addr* A variable length of a manually configured area address
- *sys_area_addr* Area address reported in a level 1 LSP received by this instance of the IS-IS protocol

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.115 `int smi_isis_get_sys_exist_state (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_exist_state)`

This call gets the state of the IS-IS router of this instance. `smi_isis_get_sys_exist_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_exist_state* State of the IS-IS router, including:
 - 1 Active
 - 2 Not in service
 - 3 Not ready
 - 4 Create and go
 - 5 Create and wait
 - 6 Destroy

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.116 `int smi_isis_get_sys_id (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_char * sys_id)`

This call gets the system ID for the instance of the IS-IS protocol. `smi_isis_get_sys_id`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_id* Pointer to the system ID string

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.117 `int smi_isis_get_sys_l2_to_l1_leaking (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_l2_to_l1_leaking)`

This call gets the state of the level 2 to level 1 route leaking for this instance of the IS-IS protocol. `smi_isis_get_sys_l2_to_l1_leaking`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_l2_to_l1_leaking* state of the level 2 to level 1 route leaking, including:
 - 1 True
 - 2 False (default)

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.118 `int smi_isis_get_sys_level_lsp_bufsize (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * sys_level_lsp_bufsize)`

This call gets the maximum size of LSPs and SNPs originated by this Intermediate System at this level. `smi_isis_get_sys_level_lsp_bufsize`

Parameters:

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

- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- *sys_level_lsp_bufsize* LSP buffer size

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.119 `int smi_isis_get_sys_level_metric_style (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * sys_level_metric_style)`

This call gets the metric style for the instance of the IS-IS protocol at this level. `smi_isis_get_sys_level_metric_style`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- *sys_level_metric_style* Metric style at this level, including:
 - 1 isisMetricStyleNarrow
 - 2 isisMetricStyleWide
 - 3 isisMetricStyleBoth

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.120 `int smi_isis_get_sys_level_min_lsp_gen_interval (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * min_lsp_gen_interval)`

This call gets Minimum interval, in seconds, between successive generation of LSPs with the same LSPID at this level by this Intermediate System. `smi_isis_get_sys_level_min_lsp_gen_interval`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0

- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- *min_lsp_gen_interval* minimum LSP generation interval

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.121 `int smi_isis_get_sys_level_overload_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * level_overload_state)`

This function gets the state of the database at this level. The value 'off' indicates that IS-IS is not active at this level. The value 'on' indicates that IS-IS is active at this level and is not overloaded. `smi_isis_get_sys_level_overload_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- *level_overload_state* system level overload state

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.122 `int smi_isis_get_sys_level_set_overload (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * sys_level_set_overload)`

This call gets the state of the overload bit for the instance of the IS-IS protocol at this level. `smi_isis_get_sys_level_set_overload`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- *sys_level_set_overload* State of the overload bit, including:
 - 1 isisTruthValueTrue
 - 2 isisTruthValueFalse

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.123 `int smi_isis_get_sys_level_set_overload_until (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t sys_level_set_overload_until)`

This call sets the time, in seconds, the overload bit should be set for the instance of the IS-IS protocol at this level. `smi_isis_get_sys_level_set_overload_until`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- *sys_level_set_overload_until* Time, in seconds, the overload bit should be set

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.124 `int smi_isis_get_sys_level_spf_considers (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * sys_level_spf_considers)`

This call gets the metric to be considered in the SPF computation for the instance of the IS-IS protocol at this level. `smi_isis_get_sys_level_spf_considers`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- *sys_level_spf_considers* Metric to be considered in the SPF computation at this level, including:
 - 1 isisMetricStyleNarrow
 - 2 isisMetricStyleWide
 - 3 isisMetricStyleBoth

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.125 `int smi_isis_get_sys_level_te_enabled (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * sys_level_te_enabled)`

This call gets the state of the traffic engineering for the instance of the IS-IS protocol at this level. `smi_isis_get_sys_level_te_enabled`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- *sys_level_te_enabled* State of the traffic engineering at this level, including:
 - 1 isisTruthValueTrue
 - 2 isisTruthValueFalse

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.126 `int smi_isis_get_sys_log_adj_changes (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_log_adj_changes)`

This call gets the state of the log generation when an IS-IS adjacency changes state (up or down). `smi_isis_get_sys_log_adj_changes`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_log_adj_changes* State of the log generation when an IS-IS adjacency changes state
 - 1 isisTruthValueTrue
 - 2 isisTruthValuefalse

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.127 `int smi_isis_get_sys_max_age (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_max_age)`

This call gets the system max age value for LSPs generated by this instance of the IS-IS protocol. `smi_isis_get_sys_max_age`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_max_age* Returns the RemainingLifeTime value of an LSP

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.128 `int smi_isis_get_sys_max_area_addrs (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_max_area_addrs)`

This call gets the maximum number of area addresses to be permitted for this instance of the IS-IS protocol. `smi_isis_get_sys_max_area_addrs`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_max_area_addrs* Maximum number of area addresses

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.129 `int smi_isis_get_sys_max_lsp_gen_interval (struct smiclient_globals
* azg, u_int32_t vr_id, u_int32_t instance, u_int32_t *
sys_max_lsp_gen_interval)`

This call gets the maximum interval, in seconds, between generated LSPs by this instance of the IS-IS protocol. `smi_isis_get_sys_max_lsp_gen_interval`

Parameters:

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

- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_max_lsp_gen_interval* Maximum interval between generated LSPs. 900 is returned by default

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.130 int smi_isis_get_sys_max_path_splits (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_max_path_splits)

This call sets the maximum number of paths with equal routing metric values permitted to split between. Only the default value can be set. smi_isis_get_sys_max_path_splits

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_max_path_splits* Maximum number of paths with equal routing metric value. Two is returned by default

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.131 int smi_isis_get_sys_next_circ_index (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_next_circ_index)

This call gets the next ISIS circ index value for this instance of the IS-IS protocol. smi_isis_get_sys_next_circ_index

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_next_circ_index* Next ISIS circ Index value

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.132 `int smi_isis_get_sys_poll_es_hello_rate (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t * poll_es_hello_rate)`

This call gets the value, in seconds, to be used for the suggested ES configuration timer in ISH PDUs when soliciting the ES configuration. `smi_isis_get_sys_poll_es_hello_rate`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *poll_es_hello_rate* Value, in seconds, to be used for the suggested ES configuration timer in ISH PDUs. 50 is returned by default

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.133 `int smi_isis_get_sys_receive_lsp_bufsize (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_receive_lsp_bufsize)`

This call gets the size of the largest buffer this instance can store. `smi_isis_get_sys_receive_lsp_bufsize`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_receive_lsp_bufsize* Size of the largest receive buffer. ISIS_PDU_MAX_LENGTH is returned by default

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.134 `int smi_isis_get_sys_stat_auth_fails (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * stat_auth_fails)`

This call gets the number of authentication failures. `smi_isis_get_sys_stat_auth_fails`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *stat_auth_fails* Integer that contains the number of authentication failures

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.135 `int smi_isis_get_sys_stat_auth_type_fails (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * stat_auth_type_fails)`

This call gets the number of authentication type mismatches. `smi_isis_get_sys_stat_auth_type_fails`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *stat_auth_type_fails* Integer that contains the number of authentication type mismatches

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.136 `int smi_isis_get_sys_stat_corrupted_lsps (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * stat_corrupted_lsps)`

This call gets the number of corrupted in-memory LSPs detected. `smi_isis_get_sys_stat_corrupted_lsps`

Parameters:

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

- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *stat_corrupted_lsps* Integer that contains number of corrupted in-memory LSPs detected

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.137 `int smi_isis_get_sys_stat_exceed_max_seqnums (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * stat_exceed_max_seqnums)`

This call gets the number of times the IS has attempted to exceed the maximum sequence number. smi_isis_get_sys_stat_exceed_max_seqnums

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *stat_exceed_max_seqnums* Integer containing the number of times IS attempted to exceed the max sequence number

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.138 `int smi_isis_get_sys_stat_id_len_mismatches (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * stat_id_len_mismatches)`

This call gets the number of times a PDU is received with a different value for ID field length to that of the receiving system. smi_isis_get_sys_stat_id_len_mismatches

Parameters:

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

- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *stat_id_len_mismatches* Integer that contains number of times a PDU is received with a different value for ID field length to that of the receiving system

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.139 `int smi_isis_get_sys_stat_lsp_purges (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * sys_stat_lsp_purges)`

This call gets the number of times a zero-aged copy of the system's own LSP is received from another node. smi_isis_get_sys_stat_lsp_purges

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *sys_stat_lsp_purges* Integer that contains number of times a zero-aged copy of the system's own LSP is received from some other node

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.140 `int smi_isis_get_sys_stat_lspdb_overloaded (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * sys_stat_lspdb_overloaded)`

This call gets the number of times the LSP database has become overloaded. smi_isis_get_sys_stat_lspdb_overloaded

Parameters:

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

- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *sys_stat_lspdb_overloaded* Integer that contains number of times the LSP database has become overloaded

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.141 `int smi_isis_get_sys_stat_man_addr_drop_area (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * stat_man_addr_drop_area)`

This call gets the number of times a manual address has been dropped from the area.
 smi_isis_get_sys_stat_man_addr_drop_area

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *stat_man_addr_drop_area* Integer that contains number of times a manual address has been dropped from the area

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.142 `int smi_isis_get_sys_stat_max_area_addr_mismatches (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * stat_max_area_addr_mismatches)`

This call gets the number of times a PDU is received with a different value for MaximumAreaAddresses from that of the receiving system. smi_isis_get_sys_stat_max_area_addr_mismatches

Parameters:

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

- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *stat_max_area_addr_mismatches* Integer that contains number of times a PDU is received with a different value for MaximumAreaAddresses from that of the receiving system

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.143 `int smi_isis_get_sys_stat_partition_changes (struct smiclient_globals *
 *azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t
 *stat_partition_changes)`

This call gets the number of times partition changes occurred. smi_isis_get_sys_stat_partition_changes

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *stat_partition_changes* Integer that contains number of times partition changes occurred

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.144 `int smi_isis_get_sys_stat_seqnum_skips (struct smiclient_globals *
 azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t *
 stat_seqnum_skips)`

This call gets the number of times a sequence number skip has occurred. smi_isis_get_sys_stat_seqnum_skips

Parameters:

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

- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *stat_seqnum_skips* Integer that contains number of times a sequence number skip has occurred

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.145 `int smi_isis_get_sys_stat_spf_runs (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t * stat_spf_runs)`

This call gets the number of times SPF ran at this level. `smi_isis_get_sys_stat_spf_runs`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- *stat_spf_runs* Integer that contains number of times SPF ran at the next level

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes
 ISIS_API_GET_ERROR

2.1.2.146 `int smi_isis_get_sys_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_type)`

This call gets the system type for the instance of the IS-IS protocol. `smi_isis_get_sys_type`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID

- *sys_type* Pointer to the version strings. One is returned by default. Values the following:
- 1 Level 1
 - 2 Level 2
 - 3 Level1 And Level 2

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.147 int smi_isis_get_sys_version (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, char * sys_version)

This call gets the version number of the IS-IS protocol that this instance implements. smi_isis_get_sys_version

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_version* Pointer to the version strings. One is returned by default

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.148 int smi_isis_get_sys_wait_time (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t * sys_wait_time)

This call gets the number of seconds to delay in waiting state before entering the on state. smi_isis_get_sys_wait_time

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- *sys_wait_time* Number of seconds to delay in waiting before the on state. 60 is returned by default

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_API_GET_ERROR

2.1.2.149 `int smi_isis_high_priority_tag_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, u_int32_t priority_tag)`

This function sets the high-priority tag. `smi_isis_high_priority_tag_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *priority_tag* IS-IS priority tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.150 `int smi_isis_high_priority_tag_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function unsets the high-priority tag. `smi_isis_high_priority_tag_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.151 `int smi_isis_hostname_dynamic_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int flag)`

This function configures the dynamic hostname TLV capability. `smi_isis_hostname_dynamic_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

- ← *flag* Method for dynamic-hostname, (0|1)
 - 0 Hostname given by router hostname command.
 - 1 Hostname given by IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.152 int smi_isis_hostname_dynamic_unset (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*)

This function unconfigures the dynamic hostname TLV capability. smi_isis_hostname_dynamic_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.153 int smi_isis_if_auth_key_chain_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *name*, char * *key_chain*, int *level*)

This function configures the key chain to be used for authentication. smi_isis_if_auth_key_chain_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *key_chain* Key chain used for authentication
- ← *level* IS-IS instance level (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 ISIS_API_SET_ERR_NO_AUTH_MD5_EXIST

**2.1.2.154 int smi_isis_if_auth_key_chain_unset (struct smiclient_globals * azg,
 u_int32_t vr_id, char * name, int level)**

This function removes the key chain to be used for authentication. smi_isis_if_auth_key_chain_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

**2.1.2.155 int smi_isis_if_auth_mode_hmac_md5_set (struct smiclient_globals *
 azg, u_int32_t vr_id, char * name, int level)**

This function sets the authentication mode to MD5. smi_isis_if_auth_mode_hmac_md5_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* interface name
- ← *level* level IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 ISIS_API_SET_ERR_AUTH_TEXT_EXIST

2.1.2.156 int smi_isis_if_auth_mode_hmac_md5_unset (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *name*, int *level*)

This call implements the no parameter of the isis authentication mode md5 command to unset the authentication mode to MD5. smi_isis_if_auth_mode_hmac_md5_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* interface name
- ← *level* level IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 ISIS_API_SET_ERR_AUTH_TEXT_EXIST

2.1.2.157 int smi_isis_if_auth_mode_text_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *name*, int *level*)

This function configures the isis authentication mode to text. smi_isis_if_auth_mode_text_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* interface name
- ← *level* level IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 ISIS_API_SET_ERR_AUTH_MD5_EXIST

**2.1.2.158 int smi_isis_if_auth_mode_text_unset (struct smiclient_globals * azg,
 u_int32_t vr_id, char * name, int level)**

This function unsets the isis authentication mode to text. smi_isis_if_auth_mode_text_unset

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router Id
- ← **name** interface name
- ← **level** level IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

**2.1.2.159 int smi_isis_if_auth_send_only_set (struct smiclient_globals * azg,
 u_int32_t vr_id, char * name, int level)**

This function configures the send-only option, that is, not to validate the authentication on the hello PDUs. smi_isis_if_auth_send_only_set

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router Id
- ← **name** Interface name
- ← **level** IS-IS instance level (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INVALID_IS_TYPE
ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.160 int smi_isis_if_auth_send_only_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name, int level)

This function unconfigure the send-only option, that is, to validate the authentication on the hello PDUs. smi_isis_if_auth_send_only_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Both Level-1, Level-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INVALID_IS_TYPE
ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.161 int smi_isis_if_circuit_type_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, int type)

This function sets the interface's circuit type. smi_isis_if_circuit_type_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *type* IS-IS Circuit-type, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_NOT_ENABLED
 ISIS_API_SET_ERR_IF_NOT_EXIST

**2.1.2.162 int smi_isis_if_circuit_type_unset (struct smiclient_globals * azg,
 u_int32_t vr_id, char * name)**

This function sets the interface's circuit type to default. smi_isis_if_circuit_type_unset

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual Router Id
 ← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_IF_NOT_ENABLED
 ISIS_API_SET_ERR_IF_NOT_EXIST
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

**2.1.2.163 int smi_isis_if_csnp_interval_set (struct smiclient_globals * azg,
 u_int32_t vr_id, char * name, u_int32_t if_csnp_interval, int level)**

This function sets the complete sequence number PDUs (CSNPs) interval for the interface. smi_isis_if_csnp_interval_set

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual Router Id
 ← *name* Interface name
 ← *if_csnp_interval* Interval in seconds. <0-65535>
 ← *level* IS-IS instance level, (1|2|3)
 1 Level-1
 2 Level-2
 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.164 `int smi_isis_if_csnp_interval_unset (struct smiclient_globals * azg,
u_int32_t vr_id, char * name, int level)`

This function unsets the complete sequence number PDUs (CSNPs) interval for the interface. `smi_isis_if_csnp_interval_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INVALID_IS_TYPE
ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.165 `int smi_isis_if_hello_interval_minimal_set (struct smiclient_globals *
azg, u_int32_t vr_id, char * name, int level)`

This function configures the Holdtime in Hello PDU to 1 second. `smi_isis_if_hello_interval_minimal_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.166 `int smi_isis_if_hello_interval_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, u_int32_t if_hello_interval, int level)`

This function configures interface's Hello interval. `smi_isis_if_hello_interval_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *if_hello_interval* Interval in seconds. <0-65535>
- ← *level* IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.167 `int smi_isis_if_hello_interval_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name, int level)`

This function unconfigure interface's Hello interval. `smi_isis_if_hello_interval_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.168 `int smi_isis_if_hello_multiplier_set (struct smiclient_globals * azg,
u_int32_t vr_id, char * name, u_int32_t multi, int level)`

This function configures the interface's Hello-Multiplier value. `smi_isis_if_hello_multiplier_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *multi* Multiplier for Hello holding time. <2-100>
- ← *level* IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INVALID_IS_TYPE
ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.169 `int smi_isis_if_hello_multiplier_unset (struct smiclient_globals * azg,
u_int32_t vr_id, char * name, int level)`

This function unconfigure the interface's Hello-Multiplier value. Default value is 3.
`smi_isis_if_hello_multiplier_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INVALID_IS_TYPE
ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.170 `int smi_isis_if_hello_padding_set (struct smiclient_globals * azg,
u_int32_t vr_id, char * name)`

This function enables IS-IS Hello packet padding. `smi_isis_if_hello_padding_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.171 `int smi_isis_if_hello_padding_unset (struct smiclient_globals * azg,
u_int32_t vr_id, char * name)`

This function disables IS-IS Hello packet padding. `smi_isis_if_hello_padding_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT

2.1.2.172 `int smi_isis_if_ip_router_set (struct smiclient_globals * azg,
u_int32_t vr_id, char * name, char * tag)`

This function enables IP router interface commands. `smi_isis_if_ip_router_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_AREA_TAG_TOO_LONG
ISIS_API_SET_ERR_AREA_TAG_NOT_MATCHED
ISIS_API_SET_ERR_IF_NOT_EXIST
ISIS_API_SET_ERR_IF_NOT_ENABLED

2.1.2.173 `int smi_isis_if_ip_router_unset (struct smiclient_globals * azg,
u_int32_t vr_id, char * name, char * tag)`

This function disable IP router interface commands. smi_isis_if_ip_router_unset

Parameters:

← *azg* Pointer to the SMI client global structure
← *vr_id* Virtual Router Id
← *name* Interface name
← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_AREA_TAG_TOO_LONG
ISIS_API_SET_ERR_AREA_TAG_NOT_MATCHED
ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.174 `int smi_isis_if_ipv6_router_set (struct smiclient_globals * azg,
u_int32_t vr_id, char * name, char * tag)`

This function enables the interface for IPv6 routing. smi_isis_if_ipv6_router_set

Parameters:

← *azg* Pointer to the SMI client global structure
← *vr_id* Virtual Router Id
← *name* Interface name
← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_AREA_TAG_TOO_LONG
ISIS_API_SET_ERR_AREA_TAG_NOT_MATCHED
ISIS_API_SET_ERR_IF_NOT_EXIST
ISIS_API_SET_ERR_IF_NOT_ENABLED

2.1.2.175 `int smi_isis_if_ipv6_router_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name, char * tag)`

This function disable IPV6 router interface commands. `smi_isis_if_ipv6_router_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_AREA_TAG_TOO_LONG
 ISIS_API_SET_ERR_AREA_TAG_NOT_MATCHED
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.176 `int smi_isis_if_lsp_interval_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, u_int32_t if_lsp_interval)`

This function configures the interface's LSP transmission interval. `smi_isis_if_lsp_interval_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *if_lsp_interval* Interval in milliseconds. <1-4294967295>

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.177 `int smi_isis_if_lsp_interval_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name)`

This function unconfigures the interface's LSP transmission interval. `smi_isis_if_lsp_interval_unset`

Parameters:

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

← *vr_id* Virtual Router Id

← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.178 int smi_isis_if_mesh_group_block_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name)

This function configures the interface as mesh-group blocked. smi_isis_if_mesh_group_block_set

Parameters:

← *azg* Pointer to the SMI client global structure

← *vr_id* Virtual Router Id

← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_IF_NOT_EXIST
ISIS_API_SET_ERR_MESH_GROUP_ID_INVALID
ISIS_API_SET_ERR_IF_NOT_ENABLED

2.1.2.179 int smi_isis_if_mesh_group_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, u_int32_t group_id)

This function configures the mesh group ID. smi_isis_if_mesh_group_set

Parameters:

← *azg* Pointer to the SMI client global structure

← *vr_id* Virtual Router Id

← *name* Interface name

← *group_id* Mesh group ID

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_IF_NOT_EXIST
ISIS_API_SET_ERR_MESH_GROUP_ID_INVALID
ISIS_API_SET_ERR_IF_NOT_ENABLED

2.1.2.180 `int smi_isis_if_mesh_group_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name, u_int32_t group_id)`

This function unconfigure the mesh group ID or mesh group blocked. `smi_isis_if_mesh_group_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *group_id* Group Id

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_IF_NOT_EXIST
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 ISIS_API_SET_ERR_IF_NOT_ENABLED

2.1.2.181 `int smi_isis_if_metric_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, u_char metric, int level)`

This function configures the interface's metric value. `smi_isis_if_metric_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *metric* Metric value.
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.182 `int smi_isis_if_metric_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name, int level)`

This function unconfigures the interface's metric value. `smi_isis_if_metric_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.183 `int smi_isis_if_network_type_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, int type)`

This function sets the IS-IS network type to either point to point or broadcast. `smi_isis_if_network_type_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *type* Interface network type,(ISIS_IFTYPE_POINTTOPOINT | ISIS_IFTYPE_BROADCAST)

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_NETWORK_TYPE
 ISIS_API_SET_ERR_IF_NOT_ENABLED
 ISIS_API_SET_ERR_IF_NOT_EXIST

2.1.2.184 `int smi_isis_if_network_type_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name)`

This function sets the IS-IS network type to the default value. `smi_isis_if_network_type_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_IF_NOT_ENABLED
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.185 `int smi_isis_if_password_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, char * passwd, int level)`

This function sets the interface's authentication password. `smi_isis_if_password_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *passwd* Authentication key, null-terminated
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_PASSWORD_TOO_LONG
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 ISIS_API_SET_ERR_AUTH_MD5_EXIST
 ISIS_API_SET_ERR_AUTH_TEXT_EXIST

2.1.2.186 `int smi_isis_if_password_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name, int level)`

This function unsets the interface's authentication password. `smi_isis_if_password_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.187 `int smi_isis_if_priority_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, u_char priority, int level)`

This function sets the interface's Priority value for Designated Router election. `smi_isis_if_priority_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *priority* Priority for Designated Router election. <0-127>
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.188 `int smi_isis_if_priority_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name, int level)`

This function unsets the interface's Priority value for Designated Router election. `smi_isis_if_priority_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.189 `int smi_isis_if_retransmit_interval_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, u_int32_t if_retransmit_interval)`

This function resets the LSP retransmission interval. `smi_isis_if_retransmit_interval_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_IF_NOT_EXIST
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 ISIS_API_SET_ERR_IF_NOT_ENABLED

2.1.2.190 `int smi_isis_if_tag_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, u_int32_t tag, int level)`

This function sets the priority tag. `smi_isis_if_tag_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.191 int smi_isis_if_tag_unset (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *name*, int *level*)

This function unsets the priority tag. smi_isis_if_tag_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.192 int smi_isis_if_wide_metric_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *name*, u_int32_t *metric*, int *level*)

This function configures the interface's wide metric value. smi_isis_if_wide_metric_set

Parameters:

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

- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *metric* Metric value.
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.193 int smi_isis_if_wide_metric_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name, int level)

This function unconfigures the interface's wide metric value. smi_isis_if_wide_metric_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.194 int smi_isis_ignore_lsp_errors_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)

This function ignores receiving LSPs(Link State Packets) with checksum error. LSP will be accepted as if it is valid. smi_isis_ignore_lsp_errors_set

Parameters:

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

- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.195 int smi_isis_ignore_lsp_errors_unset (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*)

This function validates receiving the LSP checksum. The LSP will be rejected if the checksum has an error. smi_isis_ignore_lsp_errors_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.196 int smi_isis_instance_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *name*)

This function creates an IS-IS instance for enabling a routing process. smi_isis_instance_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_AREA_TAG_TOO_LONG
ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.197 `int smi_isis_instance_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name)`

This function deletes an IS-IS instance. `smi_isis_instance_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_AREA_TAG_TOO_LONG
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.198 `int smi_isis_instance_unset_restart (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This call implements the part of the restart isis command to force shutdown of the IS-IS instance. This stores routes in the NSM, and shuts down the ISIS daemon. `smi_isis_instance_unset_restart`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* Area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_AREA_TAG_TOO_LONG

2.1.2.199 `int smi_isis_is_type_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int is_type)`

This function sets IS Level routing process as a station router only or as both a station router and an area router or as an area router only. `smi_isis_is_type_set`

Parameters:

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

- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *is_type* IS-IS instance level type (1|2|3)
 - 1 Level-1
 - 2 Level-2-only
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_L2_INSTANCE_EXIST

2.1.2.200 int smi_isis_is_type_unset (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, int *is_type*)

This function reset IS-IS Level to default. smi_isis_is_type_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *is_type* IS-IS instance level type (1|2|3)
 - 1 Level-1
 - 2 Level-2-only
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_L2_INSTANCE_EXIST

2.1.2.201 int smi_isis_ispf_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, int *ispf_level*)

This function enables incremental SPF for routing process. smi_isis_ispf_set

Parameters:

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

- ← *tag* IS-IS instance area tag
- ← *ispf_level* The iSPF level

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.202 int smi_isis_ispf_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)

This function unsets incremental SPF for routing process. smi_isis_ispf_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.203 int smi_isis_l1_snp_auth_send_only (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char * passwd)

This function sets the authentication password for the Level-1 SNP(Sequence number PDUs) but not check the password in SNP PDUs that it receives. smi_isis_l1_snp_auth_send_only

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *passwd* Authentication key, null-terminated

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_PASSWORD_TOO_LONG
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_AUTH_MD5_EXIST
 ISIS_API_SET_ERR_AUTH_TEXT_EXIST

**2.1.2.204 int smi_isis_l1_snp_auth_validate_set (struct smiclient_globals * *azg*,
u_int32_t *vr_id*, char * *tag*, char * *passwd*)**

This function sets the authentication password for the Level-1 SNP(Sequence number PDUs)and check the password in SNPs that it receives. smi_isis_l1_snp_auth_validate_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *passwd* Authentication key, null-terminated

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_PASSWORD_TOO_LONG
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
ISIS_API_SET_ERR_AUTH_MD5_EXIST
ISIS_API_SET_ERR_AUTH_TEXT_EXIST

**2.1.2.205 int smi_isis_l2_snp_auth_send_only (struct smiclient_globals * *azg*,
u_int32_t *vr_id*, char * *tag*, char * *passwd*)**

This function sets the authentication password for the Level-2 domain and SNP(Sequence number PDUs) but not check the password in SNP PDUs that it receives. smi_isis_l2_snp_auth_send_only

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *passwd* Authentication key, null-terminated

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_PASSWORD_TOO_LONG
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
ISIS_API_SET_ERR_AUTH_MD5_EXIST
ISIS_API_SET_ERR_AUTH_TEXT_EXIST

2.1.2.206 `int smi_isis_l2_snp_auth_validate_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char * passwd)`

This function sets the authentication password for the Level-2 domain and SNP(Sequence number PDUs),also checks the password in SNPs that it receives.
smi_isis_l2_snp_auth_validate_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *passwd* Authentication key, null-terminated

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_PASSWORD_TOO_LONG
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_AUTH_MD5_EXIST
 ISIS_API_SET_ERR_AUTH_TEXT_EXIST

2.1.2.207 `int smi_isis_lsp_gen_interval_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level, u_char lsp_gen_interval)`

This function configures the minimum interval between regenerating the same LSP.
smi_isis_lsp_gen_interval_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level (1|2|3)
 - 1 Level-1
 - 2 Level-2-only
 - 3 Both Level-1, Level-2
- ← *lsp_gen_interval* Interval in seconds <1-120>

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.208 `int smi_isis_lsp_gen_interval_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function resets the minimum interval between regenerating the same LSP. `smi_isis_lsp_gen_interval_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.209 `int smi_isis_lsp_mtu_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int size, int level)`

This function sets Link state Packet(lsp) MTU . `smi_isis_lsp_mtu_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *size* IS-IS size
- ← *level* IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.210 `int smi_isis_lsp_mtu_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level)`

This function restore Link state Packet(lsp) MTU to default of 1492 bytes. `smi_isis_lsp_mtu_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level, (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.211 `int smi_isis_lsp_refresh_interval_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, u_int32_t lsp_refresh_interval)`

This function sets the LSP refresh interval. `smi_isis_lsp_refresh_interval_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *lsp_refresh_interval* Interval in seconds. <1-65535>

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.212 `int smi_isis_lsp_refresh_interval_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function resets the LSP refresh interval. `smi_isis_lsp_refresh_interval_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

**2.1.2.213 int smi_isis_max_area_addr_set (struct smiclient_globals * *azg*,
u_int32_t *vr_id*, char * *tag*, u_char *limit*)**

This function sets the maximum number of ISIS areas that can be configured on a router. By default, ISIS permits a maximum of three areas that can be defined on a router. smi_isis_max_area_addr_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *limit* The maximum number of areas in the network <3-254>

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
ISIS_API_SET_ERR_INVALID_VALUE
ISIS_API_SET_ERR_MAX_AREA

**2.1.2.214 int smi_isis_max_area_addr_unset (struct smiclient_globals * *azg*,
u_int32_t *vr_id*, char * *tag*)**

This function sets the maximum number of ISIS areas to its default(3). smi_isis_max_area_addr_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.215 `int smi_isis_max_lsp_lifetime_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, u_int32_t max_lifetime)`

This function configures the maximum LSP lifetime. `smi_isis_max_lsp_lifetime_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *max_lifetime* Maximum LSP lifetime in seconds. <1-65535>

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.216 `int smi_isis_max_lsp_lifetime_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function unconfigure the maximum LSP lifetime, and set it to the default value 1200 (seconds). `smi_isis_max_lsp_lifetime_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.217 `int smi_isis_metric_style_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level)`

This function configures the metric style as wide in TLVs. `smi_isis_metric_style_set`

Parameters:

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

- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_TE_ENABLED
 ISIS_API_SET_ERR_MULTI_TOPOLOGY_ENABLED
 ISIS_API_SET_ERR_PROTOCOL_TOPOLOGY_ENABLED

2.1.2.218 int smi_isis_metric_style_transition_narrow_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, int *level*)

This function configures metric-style as transition narrow in TLVs. smi_isis_metric_style_transition_narrow_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_TE_ENABLED
 ISIS_API_SET_ERR_MULTI_TOPOLOGY_ENABLED
 ISIS_API_SET_ERR_PROTOCOL_TOPOLOGY_ENABLED

2.1.2.219 int smi_isis_metric_style_transition_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, int *level*)

This function configures the metric-style transition in TLVs. smi_isis_metric_style_transition_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_TE_ENABLED
 ISIS_API_SET_ERR_MULTI_TOPOLOGY_ENABLED
 ISIS_API_SET_ERR_PROTOCOL_TOPOLOGY_ENABLED

2.1.2.220 int smi_isis_metric_style_transition_wide_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level)

This function configures metric-style as transition wide in TLVs. smi_isis_metric_style_transition_wide_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_TE_ENABLED
 ISIS_API_SET_ERR_MULTI_TOPOLOGY_ENABLED
 ISIS_API_SET_ERR_PROTOCOL_TOPOLOGY_ENABLED

2.1.2.221 `int smi_isis_metric_style_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level)`

This function unconfigures the metric style in TLVs. `smi_isis_metric_style_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_TE_ENABLED
 ISIS_API_SET_ERR_MULTI_TOPOLOGY_ENABLED
 ISIS_API_SET_ERR_PROTOCOL_TOPOLOGY_ENABLED

2.1.2.222 `int smi_isis_mpls_traffic_eng_router_id_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, struct pal_in4_addr router_id)`

This function configures the TE router-ID. `smi_isis_mpls_traffic_eng_router_id_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *router_id* Router ID to be set

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_ROUTER_ID
 ISIS_API_SET_ERR_TE_NOT_ENABLED

2.1.2.223 `int smi_isis_mpls_traffic_eng_router_id_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function unconfigures the TE router-ID. `smi_isis_mpls_traffic_eng_router_id_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.224 `int smi_isis_mpls_traffic_eng_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level)`

This function enables traffic engineering in both level-1 and level-2 routers. `smi_isis_mpls_traffic_eng_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_WIDE_METRIC_NOT_SET

2.1.2.225 `int smi_isis_mpls_traffic_eng_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level)`

This function disables traffic engineering in both level-1 and level-2 routers. `smi_isis_mpls_traffic_eng_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_TE_NOT_ENABLED

2.1.2.226 int smi_isis_multi_topology_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, u_char *level*)

This function configures topology type as multi-topology in TLVs and SPF calculation.
 smi_isis_multi_topology_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_WIDE_METRIC_NOT_SET
 ISIS_API_SET_ERR_PROTOCOL_TOPOLOGY_SET

2.1.2.227 int smi_isis_multi_topology_transition_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, u_char *level*)

This function configures the topology type as multi-topology transition in TLVs and SPF calculation. smi_isis_multi_topology_transition_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_WIDE_METRIC_NOT_SET
 ISIS_API_SET_ERR_PROTOCOL_TOPOLOGY_SET

2.1.2.228 `int smi_isis_multi_topology_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, u_char level)`

This function configures the topology type as single-topology in TLVs and SPF calculation. `smi_isis_multi_topology_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.229 `int smi_isis_net_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char * net)`

This function configures Network Entity Title (NET) for the process. `smi_isis_net_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *net* Network entity title in string

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_NET_WRONG_FORMAT
 ISIS_API_SET_ERR_NET_INVALID_LENGTH
 ISIS_API_SET_ERR_SYSTEM_ID_CANT_CHANGED
 ISIS_API_SET_ERR_DEFAULT_TOO_MANY_AREA_ADDRESSES
 ISIS_API_SET_ERR_TOO_MANY_AREA_ADDRESSES

2.1.2.230 int smi_isis_net_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char * net)

This function unconfigures Network Entity Title (NET) for the process. smi_isis_net_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *net* Network entity title in string

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_NET_WRONG_FORMAT
 ISIS_API_SET_ERR_NET_INVALID_LENGTH
 ISIS_API_SET_ERR_SYSTEM_ID_NOT_CONFIGURED
 ISIS_API_SET_ERR_SYSTEM_ID_NOT_MATCHED

2.1.2.231 int smi_isis_no_debug (struct smiclient_globals * azg, int vr_id, int debug)

Use this function to turn off debugging for specified criteria. smi_isis_no_debug

Parameters:

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

← *vr_id* Virtual router id

← *debug* Pass debug flag as following:

- SMI_ISIS_DEBUG_ALL - Enables all debugging
- SMI_ISIS_DEBUG_IFSM - Debugging for interface finite state machine
- SMI_ISIS_DEBUG_NFSM - Debugging for neighbor finite state machine
- SMI_ISIS_DEBUG_PDU - Debugging for protocol data unit
- SMI_ISIS_DEBUG_LSP - Debugging for label switched path
- SMI_ISIS_DEBUG_SPF - Debugging for shortest path first route calculation
- SMI_ISIS_DEBUG_CHECKSUM - Debugging for checksums
- SMI_ISIS_DEBUG_AUTH - Debugging for authentication
- SMI_ISIS_DEBUG_LOCUPD - Debugging for local updates
- SMI_ISIS_DEBUG_PROTOERROR - Debugging for protocol errors
- SMI_ISIS_DEBUG_HELLO - Debugging for hello processing
- SMI_ISIS_DEBUG_EVENTS - Debugging for internal events
- SMI_ISIS_DEBUG_NSM - Debugging for NSM messages
- SMI_ISIS_DEBUG_RIB - Debugging for RIB messages
- SMI_ISIS_DEBUG_BFD - Debugging for bidirectional forwarding detection
- SMI_ISIS_DEBUG_MPLS - Debugging for multiprotocol label switching

Returns:

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

2.1.2.232 int smi_isis_parse_sys_id (struct smiclient_globals * *azg*, char * *arg*, u_char * *sys_id*)

This function parses system ID. smi_isis_parse_sys_id

Parameters:

← *azg* Pointer to the SMI client global structure

← *arg* Argument pointer

← *sys_id* Source ID

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes

ISIS_API_SET_ERR_NET_WRONG_FORMAT

ISIS_API_SET_ERR_NET_INVALID_LENGTH

2.1.2.233 `int smi_isis_passive_interface_default_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This call sets all interfaces into passive mode, except the highpriority interface. `smi_isis_passive_interface_default_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.234 `int smi_isis_passive_interface_default_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This call resets all interfaces to active mode. `smi_isis_passive_interface_default_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.235 `int smi_isis_passive_interface_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, char * name)`

This call sets the interface to passive mode for the current interface. `smi_isis_passive_interface_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*

- ← *tag* IS-IS instance area tag
- ← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_IF_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.236 int smi_isis_passive_interface_unset (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, char * *name*)

This call resets the interface to active mode for the current interface. smi_isis_passive_interface_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *tag* IS-IS instance area tag
- ← *name* Interface name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_IF_NOT_EXIST

2.1.2.237 int smi_isis_prc_interval_set (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char * *tag*, u_int32_t *min_delay*, u_int32_t *max_delay*)

This function reset parameters for Partial Route Computation (PRC). smi_isis_prc_interval_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.238 `int smi_isis_proc_clear (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This call clears isis process. smi_isis_proc_clear

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for vr_id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.239 `int smi_isis_protocol_topology_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, u_char level)`

This function enables Protocol Topology support. smi_isis_protocol_topology_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
ISIS_API_SET_ERR_VR_NOT_EXIT
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
ISIS_API_SET_ERR_INVALID_IS_TYPE
ISIS_API_SET_ERR_WIDE_METRIC_NOT_SET
ISIS_API_SET_ERR_MULTI_TOPOLOGY_SET

2.1.2.240 `int smi_isis_protocol_topology_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, u_char level)`

This function disables Protocol Topology support. smi_isis_protocol_topology_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.241 `int smi_isis_redistribute_inter_level_ipv4_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level, char * name)`

This function configures inter-level redistribution for IPv4. `smi_isis_redistribute_inter_level_ipv4_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level, (Level-1|Level-2)
- ← *name* Access-list name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.242 `int smi_isis_redistribute_inter_level_ipv4_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level)`

This function unconfigures inter-level redistribution for IPv4. `smi_isis_redistribute_inter_level_ipv4_unset`

Parameters:

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

- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level, (Level-1|Level-2)

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.243 `int smi_isis_redistribute_inter_level_ipv6_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level, char * name)`

This function configures inter-level redistribution for IPv6. `smi_isis_redistribute_inter_level_ipv6_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level, (Level-1|Level-2)
- ← *name* Access-list name

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.244 `int smi_isis_redistribute_inter_level_ipv6_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level)`

This function unconfigures inter-level redistribution for IPv6. `smi_isis_redistribute_inter_level_ipv6_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level, (Level-1|Level-2)

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.245 `int smi_isis_redistribute_ipv4_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int source, u_int32_t metric, u_char metric_type, int level, char * rmap_name)`

This function inject IPv4 routes into IS-IS from another routing protocol. `smi_isis_redistribute_ipv4_set`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual Router Id
 ← *tag* IS-IS instance area tag
 ← *source* Source of protocol, (Kernel routes|Connected routes|Static routes|RIP routes|OSPF routes|BGP routes)
 ← *metric* IS-IS metric
 ← *metric_type* External metric type, (Internal|External)
 ← *level* IS-IS instance level,(1|2|3)
 1 Level-1
 2 Level-2
 3 Level-1-2
 ← *rmap_name* Name of route-map

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_ROUTE_TYPE
 ISIS_API_SET_ERR_INVALID_METRIC_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_METRIC_VALUE
 ISIS_API_SET_ERR_WIDE_METRIC_NOT_SET

2.1.2.246 `int smi_isis_redistribute_ipv4_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int source)`

This function stop injecting IPv4 routes into IS-IS from another routing protocol. `smi_isis_redistribute_ipv4_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *source* Source of protocol, (Kernel routes|Connected routes|Static routes|RIP routes|OSPF routes|BGP routes)

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_ROUTE_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.247 `int smi_isis_redistribute_ipv6_set (struct smiclient_globals * azg,
 u_int32_t vr_id, char * tag, int source, u_int32_t metric, u_char
 metric_type, int level, char * rmap_name)`

This function inject IPv6 routes into IS-IS from another routing protocol. smi_isis_redistribute_ipv6_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *source* Source of protocol, (Kernel routes|Connected routes|Static routes|RIPng routes|OSPFv3 routes|BGP4+ routes)
- ← *metric* IS-IS metric
- ← *metric_type* External metric type, (Internal|External)
- ← *level* IS-IS instance level,(1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Level-1-2
- ← *rmap_name* Name of route-map

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIT
 ISIS_API_SET_ERR_INVALID_ROUTE_TYPE
 ISIS_API_SET_ERR_INVALID_METRIC_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_METRIC_VALUE
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.248 `int smi_isis_redistribute_ipv6_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int source)`

This function stop injecting IPv6 routes into IS-IS from another routing protocol. `smi_isis_redistribute_ipv6_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *source* Source of protocol, (Kernel routes|Connected routes|Static routes|RIPng routes| OSPFv3 routes| BGP4+ routes)

Returns:

ISIS_API_SET_SUCCESS on success, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_ROUTE_TYPE
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.249 `int smi_isis_restart_grace_period_set (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t seconds)`

This call implements the isis restart grace-period command to configure the grace period. `smi_isis_restart_grace_period_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *seconds* Grace period; the default is 65535 seconds

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_GRACE_PERIOD_INVALID
 ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.250 `int smi_isis_restart_grace_period_unset (struct smiclient_globals * azg, u_int32_t vr_id)`

This call implements the no parameter of the isis restart grace-period command to reset to the default value the grace period. `smi_isis_restart_grace_period_unset`

Parameters:

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

← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.251 `int smi_isis_restart_hello_interval_set (struct smiclient_globals * azg, u_int32_t vr_id, char * name, u_int16_t restart_hello_interval, int level)`

This call implements the isis restart-hello-interval command to configure the interval of the IS-IS Hello packet with Restart TLV. smi_isis_restart_hello_interval_set

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
 ← *name* Interface name
 ← *restart_hello_interval* Specified interval; default is 3 seconds.
 ← *level* IS-IS level

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.252 `int smi_isis_restart_hello_interval_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * name, int level)`

This call implements the no parameter of the isis restart-hello-interval command to reset the interval of the IS-IS Hello packet interval with Restart TLV to the default. smi_isis_restart_hello_interval_unset

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
 ← *name* Interface name
 ← *level* IS-IS level

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.253 `int smi_isis_restart_helper_set (struct smiclient_globals * azg, u_int32_t vr_id)`

This call implements the isis restart helper command to configure the router as the helper router. `smi_isis_restart_helper_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.254 `int smi_isis_restart_helper_unset (struct smiclient_globals * azg, u_int32_t vr_id)`

This call implements the no parameter of the isis restart helper command to unconfigure the router as the helper router. This means that a non-helper router initializes adjacency with the restarting router, and recalculates the topology. `smi_isis_restart_helper_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.255 `int smi_isis_restart_level_timer_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, u_int16_t timer, int level)`

This call implements the restart-timer command to configure the maximum timer to wait for the LSP database synchronization. `smi_isis_restart_level_timer_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* area tag
- ← *timer* Expiry timer; the default is 60 seconds
- ← *level* IS-IS level

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
ISIS_API_SET_ERR_INVALID_IS_TYPE
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.256 int smi_isis_restart_level_timer_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, int level)

This call implements the no parameter of the restart-timer command to reset the maximum timer to wait for the LSP database synchronization to the default. smi_isis_restart_level_timer_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* area tag
- ← *level* IS-IS level

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
ISIS_API_SET_ERR_INVALID_IS_TYPE
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.257 int smi_isis_restart_set (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t seconds)

This call notify NSM to restore the IS-IS routes in the NSM routing table. smi_isis_restart_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *seconds* Grace period which overrides the current grace period if the value is non-zero; the default is 65535 seconds

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_GRACE_PERIOD_INVALID

2.1.2.258 `int smi_isis_restart_suppress_adjacency_set (struct smiclient_globals * azg, u_int32_t vr_id)`

This call restarts suppress-adjacency. `smi_isis_restart_suppress_adjacency_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.259 `int smi_isis_restart_suppress_adjacency_unset (struct smiclient_globals * azg, u_int32_t vr_id)`

This call stops the suppress-adjacency. `smi_isis_restart_suppress_adjacency_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.260 `int smi_isis_set_circ_3way_enabled (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_3way_enabled)`

This call sets the status of this circuit enabled 3Way handshake. `smi_isis_set_circ_3way_enabled`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_3way_enabled* Status of the circuit enabled 3Way handshake. `isisTruthValueTrue`, `isisTruthValueFalse` (default)

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.261 `int smi_isis_set_circ_admin_state (struct smiclient_globals * azg,
 u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
 circ_admin_state)`

This call sets the administrative state of the circuit. smi_isis_set_circ_admin_state

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_admin_state* Administrative state, including:
 isisAdminStateOn (default)
 isisAdminStateOff

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.262 `int smi_isis_set_circ_exist_state (struct smiclient_globals * azg,
 u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
 circ_exist_state)`

This call sets the existence state of the circuit. smi_isis_set_circ_exist_state

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID.
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_exist_state* State of the specified circuit

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.263 `int smi_isis_set_circ_ext_domain (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_ext_domain)`

This call sets the status of the normal transmission of and interpretation of intra-domain IS-IS PDUs on this circuit. `smi_isis_set_circ_ext_domain`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_ext_domain* State of the intra-domain IS-IS PDUs, including: `isisTruthValueFalse` (default), `isisTruthValueTrue`

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
ISIS_API_SET_ERROR

2.1.2.264 `int smi_isis_set_circ_ifindex (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_ifindex)`

This call sets the value of interface index for an interface for a corresponding circuit. The interface index cannot be changed. `smi_isis_set_circ_ifindex`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0. For non-VR implementation, pass 0 for *vr_id*.
- ← *instance* Integer that contains the IS-IS instance ID.
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_ifindex* Interface index that corresponds to the circuit index

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
ISIS_API_SET_ERROR

2.1.2.265 `int smi_isis_set_circ_level (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_level)`

This call sets the type of packets that will be sent and accepted on this circuit. `smi_isis_set_circ_level`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_level* Level of the circuit, including:Level1,Level2, Level1 and Level 2

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_IF_NOT_EXIST
 ISIS_API_SET_ERR_IF_NOT_ENABLED

2.1.2.266 `int smi_isis_set_circ_level_dis_hello_timer (struct smiclient_globals *
 azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex,
 u_int32_t level, u_int32_t circ_level_dis_hello_timer)`

This call sets the period, in milliseconds, between hello PDUs on multiaccess networks when this is the designated IS. smi_isis_set_circ_level_dis_hello_timer

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index [in] level An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- ← *circ_level_dis_hello_timer* Integer that contains the hello timer of designated IS

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.267 `int smi_isis_set_circ_level_hello_multiplier (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_hello_multiplier)`

This call sets the hello multiplier which is multiplied by the corresponding HelloTimer, and the result, in seconds (rounded up), is used as the holding time in transmitted hellos, to be used by receivers of hello packets from this IS. `smi_isis_set_circ_level_hello_multiplier`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← **instance** Integer that contains the IS-IS instance ID
- ← **circindex** An integer that contains the IS-IS circuit index [in] level An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- ← **circ_level_hello_multiplier** Integer that contains the hello multiplier

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.268 `int smi_isis_set_circ_level_hello_timer (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_hello_timer)`

This call sets the maximum period, in milliseconds, between IIH PDUs on multiaccess networks at this level for LANs. The value at level 1 is used as the period between Hellos on L1L2 point-to-point circuits. `smi_isis_set_circ_level_hello_timer`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← **instance** Integer that contains the IS-IS instance ID
- ← **circindex** An integer that contains the IS-IS circuit index [in] level An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- ← **circ_level_hello_timer** Maximum period, in milliseconds, between IIH PDUs on multiaccess networks at this level for LANs. The minimum value is 1000 or 1 second. The value at level 1 is used as the period between hellos on L1 L2 point to point circuits

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.269 `int smi_isis_set_circ_level_id_octet (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_id_octet)`

This call sets a one-byte identifier that is used in protocol packets to identify a circuit for this level. The level ID octet cannot be changed. `smi_isis_set_circ_level_id_octet`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index [in] level An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- ← *circ_level_id_octet* Integer containing a 1-byte identifier that is used in protocol packets to identify a circuit

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.270 `int smi_isis_set_circ_level_lsp_throttle (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_lsp_throttle)`

This call sets minimal interval of time, in milliseconds, between transmissions of LSPs on an interface at this level. `smi_isis_set_circ_level_lsp_throttle`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID

- ← *circindex* An integer that contains the IS-IS circuit index [in] level An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- ← *circ_level_lsp_throttle* Integer that contains LSP minimum interval

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.271 `int smi_isis_set_circ_level_metric (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_metric)`

This call sets the metric value of this circuit for this level. `smi_isis_set_circ_level_metric`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for `vr_id`
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index [in] level An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- ← *circ_level_metric* Integer sub-range for default metric for single hop which picks between 0 to 63.

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.272 `int smi_isis_set_circ_level_wide_metric (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t level, u_int32_t circ_level_wide_metric)`

This call sets the wide metric value of this circuit for this level. `smi_isis_set_circ_level_wide_metric`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for `vr_id`

- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index [in] level An integer that contains the IS-IS level index, that is, 1 for level 1 IS or 2 for level 2 IS
- ← *circ_level_wide_metric* Wide metric for IS neighbors which pick between 0 to 1,677,215

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.273 `int smi_isis_set_circ_mesh_enabled (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_mesh_enabled)`

This call sets the status of the mesh group configuration of this circuit. `smi_isis_set_circ_mesh_enabled`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_mesh_enabled* To include this circuit in LSPs, even if it is not running the IS-IS protocol, including: `isisMeshGroupInactive`, `isisMeshGroupBlocked`, `isisMeshGroupSet`

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 ISIS_API_SET_ERR_IF_NOT_EXIST
 ISIS_API_SET_ERR_IF_NOT_ENABLED

2.1.2.274 `int smi_isis_set_circ_mesh_group (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_mesh_group)`

This call sets the identifier of the mesh group of this circuit. `smi_isis_set_circ_mesh_group`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_mesh_group* Integer value that represents mesh group ID

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_MESH_GROUP_ID_INVALID
 ISIS_API_SET_ERR_IF_NOT_EXIST
 ISIS_API_SET_ERR_IF_NOT_ENABLED

2.1.2.275 `int smi_isis_set_circ_passive_if (struct smiclient_globals * azg,
 u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
 circ_passive_if)`

This call sets the status to include this circuit in LSPs, even if it is not running the IS-IS protocol. smi_isis_set_circ_passive_if

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 ,For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_passive_if* To include this circuit in LSPs, even if it is not running the IS-IS protocol, including: isisTruthValueFalse (default),isisTruthValueTrue

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.276 `int smi_isis_set_circ_small_hellos (struct smiclient_globals * azg,
 u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t
 circ_small_hellos)`

This call sets the status of the IS-IS LAN hellos padding of this circuit. smi_isis_set_circ_small_hellos

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_small_hellos* Value indicates whether unpadded hellos can be sent on LAN circuits:isisTruthValueTrue , isisTruthValueFalse (default).

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.277 `int smi_isis_set_circ_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t circindex, u_int32_t circ_type)`

This call sets the type of the circuit. only broadcast and point-to-point type circuits are supported. smi_isis_set_circ_type

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *circindex* An integer that contains the IS-IS circuit index
- ← *circ_type* Type of the specified circuit

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_NETWORK_TYPE
 ISIS_API_SET_ERR_IF_NOT_EXIST
 ISIS_API_SET_ERR_IF_NOT_ENABLED

2.1.2.278 `int smi_isis_set_ip_ra_admin_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_admin_state)`

This call set the administrative state of the IP Reachable Address. smi_isis_set_ip_ra_admin_state

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- ← *ip_ra_admin_state* Administrative state of IP Reachable Address 1 On,2 Off

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.279 `int smi_isis_set_ip_ra_exist_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_exist_state)`

This call sets the state of this IP Reachable Address. smi_isis_set_ip_ra_exist_state

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- ← *ip_ra_exist_state* State of this IP reachable address

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.280 `int smi_isis_set_ip_ra_full_metric (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_full_metric)`

This call sets the wide metric value for reaching the specified destination over this circuit. `smi_isis_set_ip_ra_full_metric`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- ← *ip_ra_full_metric* Wide metric value for reaching the specified destination over this circuit

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.281 `int smi_isis_set_ip_ra_metric (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_metric)`

This call sets the metric value for reaching the specified destination over this circuit. `smi_isis_set_ip_ra_metric`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- ← *ip_ra_metric* Metric value for reaching the specified destination over this circuit

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.282 `int smi_isis_set_ip_ra_metric_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_metric_type)`

This call sets the type of metric that indicates whether the metric is internal or external.
 smi_isis_set_ip_ra_metric_type

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- ← *ip_ra_metric_type* Type of metric, including:1 Internal ,2 External

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.283 `int smi_isis_set_ip_ra_nexthop_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_nexthop_type)`

This call sets the type of the IP nexthop address. smi_isis_set_ip_ra_nexthop_type

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address

- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- ← *ip_ra_nexthop_type* Type of the IP nexthop address

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.284 `int smi_isis_set_ip_ra_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t raindex, u_int32_t type, struct prefix summ_ip_addr, u_int32_t prefixlen, u_int32_t ip_ra_type)`

This call sets the type of this IP Reachable Address. `smi_isis_set_ip_ra_type`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *instance* Integer that contains the IS-IS instance ID
- ← *raindex* Identifier to specify isisIPRAEntry
- ← *type* Type of IP Reachable Address
- ← *summ_ip_addr* Destination of IP Reachable Address
- ← *prefixlen* Length of the IP netmask of IP Reachable Address
- ← *ip_ra_type* Type of this IP Reachable Address. Those of type manual are created by the network manager. Those of type automatic are created through propagation of routing information from another routing protocol.

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.285 `int smi_isis_set_man_area_addr_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, struct smi_isis_area_addr area_addr, u_int32_t man_area_addr_state)`

This call sets the state of the manually configured area address. `smi_isis_set_man_area_addr_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0

- ← *instance* Integer that contains the IS-IS instance ID
- ← *area_addr* A variable length of a manually configured area address
- ← *man_area_addr_state* State of the manually configured area address, including:

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_AREA_TAG_TOO_LONG
 ISIS_API_SET_ERROR

2.1.2.286 `int smi_isis_set_prot_supp_exist_state (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t protocol, u_int32_t prot_supp_exist_state)`

This call gets the state of the supported protocol. `smi_isis_set_prot_supp_exist_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *protocol* An integer that contains the supported protocol, including the following values:
 - 129 ISO8473
 - 204 IP
 - 142 IPv6
- ← *prot_supp_exist_state* State of the manually configured supported protocol, including:
 - 1 Active
 - 2 NotInService
 - 3 NotReady
 - 4 CreateAndGo
 - 5 CreateAndWait
 - 6 Destroy

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.287 `int smi_isis_set_sys_admin_state (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t sys_admin_state)`

This call sets the administrative state of an instance of the IS-IS protocol. Only the default value can be set. `smi_isis_set_sys_admin_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_admin_state* Administrative state, including:
 - 1 On
 - 2 Off

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
ISIS_API_SET_ERR_AREA_TAG_TOO_LONG
ISIS_API_SET_ERROR

2.1.2.288 `int smi_isis_set_sys_exist_state (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t sys_exist_state)`

This call sets the state of the IS-IS router of this instance. `smi_isis_set_sys_exist_state`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_exist_state* State of the IS-IS router

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
ISIS_API_SET_ERR_VR_NOT_EXIST
ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
ISIS_API_SET_ERR_AREA_TAG_TOO_LONG
ISIS_API_SET_ERROR

2.1.2.289 `int smi_isis_set_sys_l2_to_l1_leaking (struct smiclient_globals * azg,
u_int32_t vr_id, u_int32_t instance, u_int32_t sys_l2_to_l1_leaking)`

This call sets the state of the level 2 to level 1 route leaking, for this instance of the IS-IS protocol. `smi_isis_set_sys_l2_to_l1_leaking`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_l2_to_l1_leaking* State of the level 2 to level 1 route leaking
 - 1 True
 - 2 False

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.290 `int smi_isis_set_sys_level_lsp_bufsize (struct smiclient_globals *
 azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t
 sys_level_lsp_bufsize)`

This call sets the maximum size of LSPs and SNPs originated by the instance of the IS-IS protocol at this level. `smi_isis_set_sys_level_lsp_bufsize`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- ← *sys_level_lsp_bufsize* Maximum size of LSPs and SNPs

Returns:

ISIS_API_GET_SUCCESS on success, otherwise following error codes ISIS_
 API_SET_ERROR

2.1.2.291 `int smi_isis_set_sys_level_set_overload (struct smiclient_globals *
 azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t
 sys_level_set_overload)`

This call sets the state of the overload bit for the instance of the IS-IS protocol at this level. `smi_isis_set_sys_level_set_overload`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID

- ← *level* An integer that contains the IS-IS level index
- ← *sys_level_set_overload* State of the overload bit, including:
 - 1 isisTruthValueTrue
 - 2 isisTruthValueFalse

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.292 `int smi_isis_set_sys_level_set_overload_until (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t sys_level_set_overload_until)`

This call sets the time, in seconds, the overload bit should be set for the instance of the IS-IS protocol at this level. `smi_isis_set_sys_level_set_overload_until`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- ← *sys_level_set_overload_until* Time, in seconds, the overload bit should be set

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_OVERLOAD_INTERVAL_INVALID

2.1.2.293 `int smi_isis_set_sys_level_spf_considers (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t sys_level_spf_considers)`

This call sets the type of metric to consider in the SPF computation for an IS-IS instance at this level. `smi_isis_set_sys_level_spf_considers`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0

- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- ← *sys_level_spf_considers* Metric type to be considered in the SPF computation at this level, including:
 - 1 isisMetricStyleNarrow
 - 2 isisMetricStyleWide
 - 3 isisMetricStyleBoth

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.294 `int smi_isis_set_sys_level_te_enabled (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t level, u_int32_t sys_level_spf_considers)`

This call sets the state of the traffic engineering for the instance of the IS-IS protocol at this level. `smi_isis_set_sys_level_te_enabled`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *level* level An integer that contains the IS-IS level index
- ← *sys_level_spf_considers* State of the traffic engineering at this level, including:
 - 1 isisTruthValueTrue
 - 2 isisTruthValueFalse

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_TE_ENABLED
 ISIS_API_SET_ERR_WIDE_METRIC_NOT_SET

2.1.2.295 `int smi_isis_set_sys_log_adj_changes (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t sys_log_adj_changes)`

This call sets the state of the log generation when an IS-IS adjacency changes state (up or down). `smi_isis_set_sys_log_adj_changes`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_log_adj_changes* State of the log generation when an IS-IS adjacency changes state, including:
 - 1 True
 - 2 False

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.296 `int smi_isis_set_sys_max_age (struct smiclient_globals * azg,
 u_int32_t vr_id, u_int32_t instance, u_int32_t sys_max_age)`

This call sets the value for the RemainingLifeTime field of the LSP, which is generated by an instance of IS-IS. `smi_isis_set_sys_max_age`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_max_age* Value to place in RemainingLifeTime field of an LSP

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERROR

2.1.2.297 `int smi_isis_set_sys_max_area_addrs (struct smiclient_globals * azg,
 u_int32_t vr_id, u_int32_t instance, u_int32_t sys_max_area_addrs)`

This call sets the maximum number of area addresses to be permitted for this instance of the IS-IS protocol. Only the default value can be set. `smi_isis_set_sys_max_area_addrs`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0

- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_max_area_addrs* Maximum number of area addresses

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE
 ISIS_API_SET_ERR_TOO_MANY_AREA_ADDRESSES
 ISIS_API_SET_ERROR

2.1.2.298 `int smi_isis_set_sys_max_lsp_gen_interval (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t sys_max_lsp_gen_interval)`

This call sets the maximum interval, in seconds, between generated LSPs by this instance of the IS-IS protocol. Only the default value can be set. `smi_isis_set_sys_max_lsp_gen_interval`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_max_lsp_gen_interval* Maximum interval between generated LSPs

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.299 `int smi_isis_set_sys_max_path_splits (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t sys_max_path_splits)`

This call sets the maximum number of paths with equal routing metric values permitted to split between. Only the default value can be set. `smi_isis_set_sys_max_path_splits`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_max_path_splits* Maximum number of paths with equal routing metric value

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.300 `int smi_isis_set_sys_poll_es_hello_rate (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t sys_poll_es_hello_rate)`

This call sets the value, in seconds, to be used for the suggested ES configuration timer in ISH PDUs when soliciting the ES configuration. Only the default value can be set. `smi_isis_set_sys_poll_es_hello_rate`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_poll_es_hello_rate* Value, in seconds, to be used for the suggested ES configuration timer in ISH PDUs

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.301 `int smi_isis_set_sys_receive_lsp_bufsize (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t sys_receive_lsp_bufsize)`

This call sets the size of the largest buffer this instance can store. `smi_isis_set_sys_receive_lsp_bufsize`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_receive_lsp_bufsize* Size of the largest receive buffer. only ISIS_PDU_MAX_LENGTH can be set

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.302 `int smi_isis_set_sys_type (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t sys_type)`

This call sets the system type for the instance of the IS-IS protocol. `smi_isis_set_sys_type`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_type* Integer value that represents type of IS-IS instance. Values include the following:
 - 1 level1IS
 - 2 level2IS
 - 3 level1L2IS

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_L2_INSTANCE_EXIST
 ISIS_API_SET_ERROR

2.1.2.303 `int smi_isis_set_sys_wait_time (struct smiclient_globals * azg, u_int32_t vr_id, u_int32_t instance, u_int32_t val)`

This call sets the seconds to delay in waiting state before entering an on state. Only the default value can be set. `smi_isis_set_sys_wait_time`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *instance* Integer that contains the IS-IS instance ID
- ← *sys_wait_time* Number of seconds to delay in waiting state before on state

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERROR

2.1.2.304 `int smi_isis_show_clns_if_nbr_api (struct smiclient_globals * azg, int vr_id, int start_index, int end_index, char * tag, char * ifname, struct list * isis_clns_nbr, u_int32_t(*) (struct list * isis_clns_nbr) callbackFunc)`

This function retrieves detailed ISIS neighbors information. `smi_isis_show_clns_if_nbr_api`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag*- tag name to retrieve the details of a ISIS neighbors attached to given tag, NULL to get the details of all ISIS neighbors
- ← *ifname*- interface name to retrieve the details of a ISIS neighbors attached to given interface, NULL to get the details of ISIS neighbors attached to all the interfaces
- ← *start_index* Start Index in case information needs to be retrieved for from begin to end indices
- ← *end_index* End Index in case information needs to be retrieved from start_index to end_index
- *isis_clns_nbr* Linked list of isis_neighbors structure
- ← *funcpointer* Pointer to a function which needs to be invoked when response is available from SMI server

Returns:

0 in case of success, otherwise one of the following error codes ISIS_DOWN

2.1.2.305 `int smi_isis_show_clns_nbr_api (struct smiclient_globals * azg, int vr_id, int start_index, int end_index, char * tag, struct list * isis_clns_nbr, u_int32_t(*) (struct list * isis_clns_nbr) callbackFunc)`

This function retrieves detailed ISIS is-neighbors information. smi_isis_show_clns_nbr_api

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag*- tag name to retrieve the details of a ISIS is-neighbors attached to given tag, NULL to get the details of all ISIS is-neighbors
- ← *start_index* Start Index in case information needs to be retrieved for from begin to end indices
- ← *end_index* End Index in case information needs to be retrieved from start_index to end_index
- *isis_clns_nbr* Linked list of isis_neighbors structure
- ← *funcpointer* Pointer to a function which needs to be invoked when response is available from SMI server

Returns:

0 in case of success, otherwise one of the following error codes ISIS_DOWN

2.1.2.306 `int smi_isis_show_clns_neighbors_api (struct smiclient_globals * azg, int vr_id, int start_index, int end_index, char * tag, struct list * isis_clns_nbr, u_int32_t (*)(struct list * isis_clns_nbr) callbackFunc)`

This function retrieves detailed ISIS neighbors information. `smi_isis_show_clns_neighbors_api`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router Id
- ← **tag**- tag name to retrieve the details of a ISIS neighbors attached to given tag, NULL to get the details of all ISIS neighbors
- ← **start_index** Start Index in case information needs to be retrieved for from begin to end indices
- ← **end_index** End Index in case information needs to be retrieved from start_index to end_index
- **isis_clns_nbr** Linked list of isis_neighbors structure
- ← **funcpointer** Pointer to a function which needs to be invoked when response is available from SMI server

Returns:

0 in case of success, otherwise one of the following error codes ISIS_DOWN

2.1.2.307 `int smi_isis_show_database (struct smiclient_globals * azg, int vr_id, int start_index, int end_index, char * flag, struct list * isisOuList, u_int32_t (*)(struct list * isisOutList) callbackFunc)`

This call displays the entire ISIS database. `smi_isis_show_database`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← **flag** - flag (detail or verbose)
- **isisOuList** Pointer to linked list of structure isisAreaEntry
- **callbackFunc** Callback function

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.308 `int smi_isis_show_database_filtered (struct smiclient_globals * azg, int vr_id, char * tag, char * lspid, char * level, char * flag, struct list * isisOutList, u_int32_t(*) (struct list * isisOutList) callbackFunc)`

This call displays the filtered ISIS database. smi_isis_show_database_filtered

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for vr_id
- ← *tag* - Area tag
- ← *lspid* - lspid
- ← *level* - level-1 or level-2
- ← *flag* - flag (detail or verbose)
- *isisOutList* Pointer to linked list of structure isisAreaEntry
- *callbackFunc* Callback function

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.309 `int smi_isis_show_global_stat (struct smiclient_globals * azg, int vr_id, int start_index, int end_index, struct list * isisOutList, u_int32_t(*) (struct list * isisOutList) callbackFunc)`

This call fetchs the isis global counters. smi_isis_show_global_stat

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for vr_id
- ← *start_index* Start Index in case information needs to be retrieved for from begin to end indices
- ← *end_index* End Index in case information needs to be retrieved from start_index to end_index
- *isisOutList* Linked list of smi_isis_global_stat structure
- ← *callbackFunc* Pointer to a function which needs to be invoked when response is available from SMI serve

Returns:

SMI_SUCCESS on success, otherwise following error codes
 SMI_ERROR
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_VALUE ISIS_API_SET_ERR_IF_NOT_EXIST
 ISIS_INVALID_INPUT_PARAM

2.1.2.310 `int smi_isis_show_if_stat (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname, struct smi_isis_if_stat * isis_if_stat, u_int32_t(*)(struct smi_isis_if_stat *isis_is_stat) callbackFunc)`

This call fetchs the isis interface counters of the given interface. `smi_isis_show_if_stat`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for vr_id
- ← **ifname** interface name
- **isis_if_stat** handle to fetch data, of type `smi_isis_if_stat`
- ← **callbackFunc** Pointer to a function which needs to be invoked when response is available from SMI serve

Returns:

SMI_SUCCESS on success, otherwise following error codes
 SMI_ERROR
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST ISIS_API_SET_ERR_VR_-
 NOT_EXIST ISIS_API_SET_ERR_IF_NOT_EXIST ISIS_INVALID_INPUT_-
 PARAM

2.1.2.311 `int smi_isis_show_tag_global_stat (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, struct smi_isis_global_stat * isis_global_stat, u_int32_t(*)(struct smi_isis_global_stat *isis_global_stat) callbackFunc)`

This call fetchs the isis global counters of the given tag. `smi_isis_show_tag_global_stat`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for vr_id
- ← **tag** IS-IS instance tag
- **isis_global_stat** handle to fetch data, of type `smi_isis_global_stat`
- ← **callbackFunc** Pointer to a function which needs to be invoked when response is available from SMI serve

Returns:

SMI_SUCCESS on success, otherwise following error codes
 SMI_ERROR
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST ISIS_API_SET_ERR_VR_-
 NOT_EXIST ISIS_API_SET_ERR_IF_NOT_EXIST ISIS_INVALID_INPUT_-
 PARAM

2.1.2.312 `int smi_isis_show_tag_if_stat (struct smiclient_globals * azg,
u_int32_t vr_id, char * ifname, char * tag, struct smi_isis_if_stat
* isis_if_stat, u_int32_t(*) (struct smi_isis_if_stat *isis_is_stat)
callbackFunc)`

This call fetchs the isis interface counters of the given interface and tag. `smi_isis_show_tag_if_stat`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for *vr_id*
- ← *ifname* interface name
- ← *tag* IS-IS instance tag
- *isis_if_stat* handle to fetch data, of type `smi_isis_if_stat`

Returns:

SMI_SUCCESS on success, otherwise following error codes

Parameters:

- ← *callbackFunc* Pointer to a function which needs to be invoked when response is available from SMI serve
- SMI_ERROR
- ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
- ISIS_API_SET_ERR_VR_NOT_EXIST
- ISIS_API_SET_ERR_IF_NOT_EXIST
- ISIS_INVALID_INPUT_PARAM

2.1.2.313 `int smi_isis_show_topology_all (struct smiclient_globals * azg,
u_int32_t vr_id, int pindex, char * tag, int level, struct list *
isisOutlist, u_int32_t(*) (struct list *isisOutlist) callbackFunc)`

This call displays ISIS ipv6 protocol related information. `smi_isis_api_show_ipv6_protocols`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0 For non-VR implementation, pass 0 for *vr_id*
- ← *start_index*
- ← *end* index
- *isisOuList* Pointer to linked list of structure `isis_infolist`
- *callbackFunc* Callback function

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.314 `int smi_isis_spf_interval_set (struct smiclient_globals * azg,
u_int32_t vr_id, char * tag, int level, u_int32_t start_delay, u_int32_t
min_delay, u_int32_t max_delay)`

This function configures the minimum and maximum interval between SPF calculations. `smi_isis_spf_interval_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag
- ← *level* IS-IS instance level (1|2|3)
 - 1 Level-1
 - 2 Level-2
 - 3 Both Level-1, Level-2
- ← *start_delay* Initial SPF delay time in milliseconds:
- ← *min_delay* Minimum delay between receiving a change to SPF calculation in milliseconds <0-2147483647>
- ← *max_delay* Maximum delay between receiving a change to SPF calculation in milliseconds <0-2147483647>

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE
 ISIS_API_SET_ERR_INVALID_VALUE

2.1.2.315 `int smi_isis_spf_interval_unset (struct smiclient_globals * azg,
u_int32_t vr_id, char * tag)`

This function unconfigure the minimum interval between SPF calculations. Default is 10 (seconds). `smi_isis_spf_interval_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* IS-IS instance area tag

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.316 `int smi_isis_summary_address_set (struct smiclient_globals * azg,
u_int32_t vr_id, char * tag, struct pal_in4_addr addr, u_char
masklen, int level, u_int8_t metric)`

This call implements the summary-address command to summarize specific IPv4 reachability information. smi_isis_summary_address_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for vr_id
- ← *tag* IS-IS instance tag
- ← *addr* IPv4 network address
- ← *masklen* Mask length
- ← *level* IS-IS instance level, including: Level-1 , Level-2, Both Level-1, Level-2
- ← *metric* Metric value

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST
 ISIS_API_SET_ERR_INVALID_IS_TYPE

2.1.2.317 `int smi_isis_summary_address_unset (struct smiclient_globals *
azg, u_int32_t vr_id, char * tag, struct pal_in4_addr addr, u_char
masklen)`

This call implements the no parameter of the summary-address command to remove the summary. smi_isis_summary_address_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID. The default value is 0, For non-VR implementation, pass 0 for vr_id
- ← *tag* IS-IS instance tag
- ← *addr* IPv4 network address
- ← *masklen* Mask length

Returns:

ISIS_API_SET_SUCCESS on success, otherwise following error codes
 ISIS_API_SET_ERR_VR_NOT_EXIST
 ISIS_API_SET_ERR_INSTANCE_NOT_EXIST

2.1.2.318 `int smi_show_ip_isis_route (struct smiclient_globals * azg, u_int32_t vr_id, struct list * outputList, u_int32_t(*) (struct list * outputList) callbackFunc)`

This function displays the isis IP Route Info. `smi_show_ip_isis_route`

Parameters:

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

Returns:

ISIS_API_SET_SUCCESS, otherwise it returns error code
ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.319 `int smi_show_ipv6_isis_route (struct smiclient_globals * azg, u_int32_t vr_id, struct list * outputList, u_int32_t(*) (struct list * outputList) callbackFunc)`

This function displays the isis IPV6 Route Info. `smi_show_ipv6_isis_route`

Parameters:

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

Returns:

ISIS_API_SET_SUCCESS, otherwise it returns error code
ISIS_API_SET_ERR_VR_NOT_EXIST

2.1.2.320 `int smi_show_isis_interface (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname, int start_index, int end_index, struct list * ifBriefList, int(*) (struct list * ifBriefList) funpointer)`

This function retrieves all interface details and brief details * * * `smi_show_isis_interface` * *

Parameters:

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

*

Parameters:

- ← *ifname*- interface name or null: for all interfaces

Returns:

ISIS_API_SET_SUCCESS, otherwise one of the following error codes
MEM_ERROR
ISIS_DOWN

Index

- smi_isis.h, [3](#)
 - smi_isis_address_family_ipv6_-unicast_unset, [36](#)
 - smi_isis_adjacency_check_ipv4_set, [36](#)
 - smi_isis_adjacency_check_ipv4_-unset, [37](#)
 - smi_isis_adjacency_check_ipv6_set, [37](#)
 - smi_isis_adjacency_check_ipv6_-unset, [37](#)
 - smi_isis_api_show_ipv6_protocols, [38](#)
 - smi_isis_area_password_set, [38](#)
 - smi_isis_area_password_unset, [39](#)
 - smi_isis_auth_key_chain_set, [39](#)
 - smi_isis_auth_key_chain_unset, [40](#)
 - smi_isis_auth_mode_hmac_md5_-set, [40](#)
 - smi_isis_auth_mode_hmac_md5_-unset, [41](#)
 - smi_isis_auth_mode_text_set, [41](#)
 - smi_isis_auth_mode_text_unset, [42](#)
 - smi_isis_auth_send_only_set, [42](#)
 - smi_isis_auth_send_only_unset, [43](#)
 - smi_isis_clear_counters, [43](#)
 - smi_isis_clear_interface_counters, [43](#)
 - smi_isis_clear_ip_route, [44](#)
 - smi_isis_clear_ipv6_route, [44](#)
 - smi_isis_cspf_set, [45](#)
 - smi_isis_cspf_unset, [45](#)
 - smi_isis_debug, [45](#)
 - smi_isis_default_information_-originate_ipv4_set, [46](#)
 - smi_isis_default_information_-originate_ipv4_unset, [47](#)
 - smi_isis_default_information_-originate_ipv6_set, [47](#)
 - smi_isis_default_information_-originate_ipv6_unset, [47](#)
 - smi_isis_distance_ipv6_set, [48](#)
 - smi_isis_distance_ipv6_unset, [48](#)
 - smi_isis_distance_set, [49](#)
 - smi_isis_distance_source_set, [49](#)
 - smi_isis_distance_source_unset, [50](#)
 - smi_isis_distance_unset, [50](#)
 - smi_isis_domain_password_set, [51](#)
 - smi_isis_domain_password_unset, [51](#)
 - smi_isis_get_circ_3way_enabled, [51](#)
 - smi_isis_get_circ_adj_changes, [52](#)
 - smi_isis_get_circ_admin_state, [52](#)
 - smi_isis_get_circ_auth_fails, [53](#)
 - smi_isis_get_circ_auth_type_fails, [53](#)
 - smi_isis_get_circ_exist_state, [54](#)
 - smi_isis_get_circ_ext_domain, [54](#)
 - smi_isis_get_circ_id_len_-mismatches, [55](#)
 - smi_isis_get_circ_ifindex, [55](#)
 - smi_isis_get_circ_init_fails, [56](#)
 - smi_isis_get_circ_lan_dis_changes, [56](#)
 - smi_isis_get_circ_level, [57](#)
 - smi_isis_get_circ_level_csnp_-interval, [57](#)
 - smi_isis_get_circ_level_dis, [58](#)
 - smi_isis_get_circ_level_dis_hello_-timer, [58](#)
 - smi_isis_get_circ_level_hello_-multiplier, [59](#)
 - smi_isis_get_circ_level_hello_-timer, [59](#)
 - smi_isis_get_circ_level_id, [60](#)
 - smi_isis_get_circ_level_id_octet, [60](#)
 - smi_isis_get_circ_level_lsp_throttle, [61](#)
 - smi_isis_get_circ_level_metric, [61](#)
 - smi_isis_get_circ_level_min_lsp_-retrans, [62](#)
 - smi_isis_get_circ_level_priority, [62](#)

- smi_isis_get_circ_level_psnp_-
interval, [63](#)
- smi_isis_get_circ_level_wide_-
metric, [63](#)
- smi_isis_get_circ_max_area_addr_-
mismatches, [64](#)
- smi_isis_get_circ_mesh_enabled, [64](#)
- smi_isis_get_circ_mesh_group, [65](#)
- smi_isis_get_circ_num_adj, [65](#)
- smi_isis_get_circ_passive_if, [66](#)
- smi_isis_get_circ_rej_adj, [66](#)
- smi_isis_get_circ_small_hellos, [67](#)
- smi_isis_get_circ_type, [67](#)
- smi_isis_get_circ_uptime, [68](#)
- smi_isis_get_ip_ra_admin_state, [68](#)
- smi_isis_get_ip_ra_exist_state, [69](#)
- smi_isis_get_ip_ra_full_metric, [69](#)
- smi_isis_get_ip_ra_metric, [70](#)
- smi_isis_get_ip_ra_metric_type, [71](#)
- smi_isis_get_ip_ra_snpa_address,
[71](#)
- smi_isis_get_ip_ra_source_type, [72](#)
- smi_isis_get_ip_ra_type, [72](#)
- smi_isis_get_is_adj_3way_state, [73](#)
- smi_isis_get_is_adj_area_address,
[73](#)
- smi_isis_get_is_adj_extended_-
circ_id, [74](#)
- smi_isis_get_is_adj_hold_time, [74](#)
- smi_isis_get_is_adj_ip_addr_type,
[75](#)
- smi_isis_get_is_adj_ip_address, [75](#)
- smi_isis_get_is_adj_nbr_priority, [76](#)
- smi_isis_get_is_adj_nbr_snpa_addr,
[76](#)
- smi_isis_get_is_adj_nbr_sys_id, [77](#)
- smi_isis_get_is_adj_nbr_sys_type,
[77](#)
- smi_isis_get_is_adj_prot_supp_-
protocol, [78](#)
- smi_isis_get_is_adj_state, [78](#)
- smi_isis_get_is_adj_uptime, [79](#)
- smi_isis_get_is_adj_usage, [79](#)
- smi_isis_get_lsp_attributes, [80](#)
- smi_isis_get_lsp_checksum, [80](#)
- smi_isis_get_lsp_lifetime_remain,
[81](#)
- smi_isis_get_lsp_pdu_length, [81](#)
- smi_isis_get_lsp_seq, [82](#)
- smi_isis_get_lsp_tlv_checksum, [82](#)
- smi_isis_get_lsp_tlv_index, [83](#)
- smi_isis_get_lsp_tlv_len, [83](#)
- smi_isis_get_lsp_tlv_seq, [84](#)
- smi_isis_get_lsp_tlv_type, [84](#)
- smi_isis_get_lsp_zero_life, [85](#)
- smi_isis_get_man_area_addr_state,
[85](#)
- smi_isis_get_packet_count_csnp, [86](#)
- smi_isis_get_packet_count_hello,
[87](#)
- smi_isis_get_packet_count_lsp, [87](#)
- smi_isis_get_packet_count_psnp, [88](#)
- smi_isis_get_packet_count_-
unknown, [88](#)
- smi_isis_get_prot_supp_exist_state,
[89](#)
- smi_isis_get_summ_addr_full_-
metric, [89](#)
- smi_isis_get_summ_addr_metric, [90](#)
- smi_isis_get_summ_addr_state, [90](#)
- smi_isis_get_sys_admin_state, [91](#)
- smi_isis_get_sys_area_addr, [91](#)
- smi_isis_get_sys_exist_state, [92](#)
- smi_isis_get_sys_id, [92](#)
- smi_isis_get_sys_l2_to_l1_leaking,
[93](#)
- smi_isis_get_sys_level_lsp_bufsize,
[93](#)
- smi_isis_get_sys_level_metric_-
style, [94](#)
- smi_isis_get_sys_level_min_lsp_-
gen_interval, [94](#)
- smi_isis_get_sys_level_overload_-
state, [95](#)
- smi_isis_get_sys_level_set_-
overload, [95](#)
- smi_isis_get_sys_level_set_-
overload_until, [96](#)
- smi_isis_get_sys_level_spf_-
considers, [96](#)
- smi_isis_get_sys_level_te_enabled,
[96](#)
- smi_isis_get_sys_log_adj_changes,
[97](#)
- smi_isis_get_sys_max_age, [97](#)
- smi_isis_get_sys_max_area_addrs,
[98](#)
- smi_isis_get_sys_max_lsp_gen_-
interval, [98](#)

- smi_isis_get_sys_max_path_splits, 99
- smi_isis_get_sys_next_circ_index, 99
- smi_isis_get_sys_poll_es_hello_rate, 99
- smi_isis_get_sys_receive_lsp_bufsize, 100
- smi_isis_get_sys_stat_auth_fails, 100
- smi_isis_get_sys_stat_auth_type_fails, 101
- smi_isis_get_sys_stat_corrupted_lsps, 101
- smi_isis_get_sys_stat_exceed_max_seqnums, 102
- smi_isis_get_sys_stat_id_len_mismatches, 102
- smi_isis_get_sys_stat_lsp_purges, 103
- smi_isis_get_sys_stat_lspdb_overloaded, 103
- smi_isis_get_sys_stat_man_addr_drop_area, 104
- smi_isis_get_sys_stat_max_area_addr_mismatches, 104
- smi_isis_get_sys_stat_partition_changes, 105
- smi_isis_get_sys_stat_seqnum_skips, 105
- smi_isis_get_sys_stat_spf_runs, 106
- smi_isis_get_sys_type, 106
- smi_isis_get_sys_version, 107
- smi_isis_get_sys_wait_time, 107
- smi_isis_high_priority_tag_set, 107
- smi_isis_high_priority_tag_unset, 108
- smi_isis_hostname_dynamic_set, 108
- smi_isis_hostname_dynamic_unset, 109
- smi_isis_if_auth_key_chain_set, 109
- smi_isis_if_auth_key_chain_unset, 110
- smi_isis_if_auth_mode_hmac_md5_set, 110
- smi_isis_if_auth_mode_hmac_md5_unset, 111
- smi_isis_if_auth_mode_text_set, 111
- smi_isis_if_auth_mode_text_unset, 112
- smi_isis_if_auth_send_only_set, 112
- smi_isis_if_auth_send_only_unset, 113
- smi_isis_if_circuit_type_set, 113
- smi_isis_if_circuit_type_unset, 114
- smi_isis_if_csnp_interval_set, 114
- smi_isis_if_csnp_interval_unset, 114
- smi_isis_if_hello_interval_minimal_set, 115
- smi_isis_if_hello_interval_set, 115
- smi_isis_if_hello_interval_unset, 116
- smi_isis_if_hello_multiplier_set, 116
- smi_isis_if_hello_multiplier_unset, 117
- smi_isis_if_hello_padding_set, 117
- smi_isis_if_hello_padding_unset, 118
- smi_isis_if_ip_router_set, 118
- smi_isis_if_ip_router_unset, 119
- smi_isis_if_ipv6_router_set, 119
- smi_isis_if_ipv6_router_unset, 119
- smi_isis_if_lsp_interval_set, 120
- smi_isis_if_lsp_interval_unset, 120
- smi_isis_if_mesh_group_block_set, 121
- smi_isis_if_mesh_group_set, 121
- smi_isis_if_mesh_group_unset, 121
- smi_isis_if_metric_set, 122
- smi_isis_if_metric_unset, 122
- smi_isis_if_network_type_set, 123
- smi_isis_if_network_type_unset, 123
- smi_isis_if_password_set, 124
- smi_isis_if_password_unset, 124
- smi_isis_if_priority_set, 125
- smi_isis_if_priority_unset, 125
- smi_isis_if_retransmit_interval_set, 126
- smi_isis_if_tag_set, 126
- smi_isis_if_tag_unset, 127
- smi_isis_if_wide_metric_set, 127
- smi_isis_if_wide_metric_unset, 128

- smi_isis_ignore_lsp_errors_set, [128](#)
- smi_isis_ignore_lsp_errors_unset, [129](#)
- smi_isis_instance_set, [129](#)
- smi_isis_instance_unset, [129](#)
- smi_isis_instance_unset_restart, [130](#)
- smi_isis_is_type_set, [130](#)
- smi_isis_is_type_unset, [131](#)
- smi_isis_ispf_set, [131](#)
- smi_isis_ispf_unset, [132](#)
- smi_isis_l1_snp_auth_send_only, [132](#)
- smi_isis_l1_snp_auth_validate_set, [132](#)
- smi_isis_l2_snp_auth_send_only, [133](#)
- smi_isis_l2_snp_auth_validate_set, [133](#)
- smi_isis_lsp_gen_interval_set, [134](#)
- smi_isis_lsp_gen_interval_unset, [134](#)
- smi_isis_lsp_mtu_set, [135](#)
- smi_isis_lsp_mtu_unset, [135](#)
- smi_isis_lsp_refresh_interval_set, [136](#)
- smi_isis_lsp_refresh_interval_unset, [136](#)
- smi_isis_max_area_addr_set, [137](#)
- smi_isis_max_area_addr_unset, [137](#)
- smi_isis_max_lsp_lifetime_set, [137](#)
- smi_isis_max_lsp_lifetime_unset, [138](#)
- smi_isis_metric_style_set, [138](#)
- smi_isis_metric_style_transition_narrow_set, [139](#)
- smi_isis_metric_style_transition_set, [139](#)
- smi_isis_metric_style_transition_wide_set, [140](#)
- smi_isis_metric_style_unset, [140](#)
- smi_isis_mpls_traffic_eng_router_id_set, [141](#)
- smi_isis_mpls_traffic_eng_router_id_unset, [141](#)
- smi_isis_mpls_traffic_eng_set, [142](#)
- smi_isis_mpls_traffic_eng_unset, [142](#)
- smi_isis_multi_topology_set, [143](#)
- smi_isis_multi_topology_transition_set, [143](#)
- smi_isis_multi_topology_unset, [144](#)
- smi_isis_net_set, [144](#)
- smi_isis_net_unset, [145](#)
- smi_isis_no_debug, [145](#)
- smi_isis_parse_sys_id, [146](#)
- smi_isis_passive_interface_default_set, [146](#)
- smi_isis_passive_interface_default_unset, [147](#)
- smi_isis_passive_interface_set, [147](#)
- smi_isis_passive_interface_unset, [148](#)
- smi_isis_prc_interval_set, [148](#)
- smi_isis_proc_clear, [148](#)
- smi_isis_protocol_topology_set, [149](#)
- smi_isis_protocol_topology_unset, [149](#)
- smi_isis_redistribute_inter_level_ipv4_set, [150](#)
- smi_isis_redistribute_inter_level_ipv4_unset, [150](#)
- smi_isis_redistribute_inter_level_ipv6_set, [151](#)
- smi_isis_redistribute_inter_level_ipv6_unset, [151](#)
- smi_isis_redistribute_ipv4_set, [152](#)
- smi_isis_redistribute_ipv4_unset, [152](#)
- smi_isis_redistribute_ipv6_set, [153](#)
- smi_isis_redistribute_ipv6_unset, [153](#)
- smi_isis_restart_grace_period_set, [154](#)
- smi_isis_restart_grace_period_unset, [154](#)
- smi_isis_restart_hello_interval_set, [155](#)
- smi_isis_restart_hello_interval_unset, [155](#)
- smi_isis_restart_helper_set, [155](#)
- smi_isis_restart_helper_unset, [156](#)
- smi_isis_restart_level_timer_set, [156](#)
- smi_isis_restart_level_timer_unset, [157](#)
- smi_isis_restart_set, [157](#)
- smi_isis_restart_suppress_adjacency_set, [157](#)
- smi_isis_restart_suppress_adjacency_unset, [158](#)

- smi_isis_set_circ_3way_enabled, 158
- smi_isis_set_circ_admin_state, 159
- smi_isis_set_circ_exist_state, 159
- smi_isis_set_circ_ext_domain, 159
- smi_isis_set_circ_ifindex, 160
- smi_isis_set_circ_level, 160
- smi_isis_set_circ_level_dis_hello_timer, 161
- smi_isis_set_circ_level_hello_multiplier, 161
- smi_isis_set_circ_level_hello_timer, 162
- smi_isis_set_circ_level_id_octet, 163
- smi_isis_set_circ_level_lsp_throttle, 163
- smi_isis_set_circ_level_metric, 164
- smi_isis_set_circ_level_wide_metric, 164
- smi_isis_set_circ_mesh_enabled, 165
- smi_isis_set_circ_mesh_group, 165
- smi_isis_set_circ_passive_if, 166
- smi_isis_set_circ_small_hellos, 166
- smi_isis_set_circ_type, 167
- smi_isis_set_ip_ra_admin_state, 167
- smi_isis_set_ip_ra_exist_state, 168
- smi_isis_set_ip_ra_full_metric, 168
- smi_isis_set_ip_ra_metric, 169
- smi_isis_set_ip_ra_metric_type, 170
- smi_isis_set_ip_ra_nexthop_type, 170
- smi_isis_set_ip_ra_type, 171
- smi_isis_set_man_area_addr_state, 171
- smi_isis_set_prot_supp_exist_state, 172
- smi_isis_set_sys_admin_state, 172
- smi_isis_set_sys_exist_state, 173
- smi_isis_set_sys_l2_to_l1_leaking, 173
- smi_isis_set_sys_level_lsp_bufsize, 174
- smi_isis_set_sys_level_set_overload, 174
- smi_isis_set_sys_level_set_overload_until, 175
- smi_isis_set_sys_level_spf_considers, 175
- smi_isis_set_sys_level_te_enabled, 176
- smi_isis_set_sys_log_adj_changes, 176
- smi_isis_set_sys_max_age, 177
- smi_isis_set_sys_max_area_addrs, 177
- smi_isis_set_sys_max_lsp_gen_interval, 178
- smi_isis_set_sys_max_path_splits, 178
- smi_isis_set_sys_poll_es_hello_rate, 178
- smi_isis_set_sys_receive_lsp_bufsize, 179
- smi_isis_set_sys_type, 179
- smi_isis_set_sys_wait_time, 180
- smi_isis_show_clns_if_nbr_api, 180
- smi_isis_show_clns_nbr_api, 181
- smi_isis_show_clns_neighbors_api, 181
- smi_isis_show_database, 182
- smi_isis_show_database_filtered, 182
- smi_isis_show_global_stat, 183
- smi_isis_show_if_stat, 183
- smi_isis_show_tag_global_stat, 184
- smi_isis_show_tag_if_stat, 184
- smi_isis_show_topology_all, 185
- smi_isis_spf_interval_set, 185
- smi_isis_spf_interval_unset, 186
- smi_isis_summary_address_set, 186
- smi_isis_summary_address_unset, 187
- smi_show_ip_isis_route, 187
- smi_show_ipv6_isis_route, 188
- smi_show_isis_interface, 188
- smi_isis_address_family_ipv6_unicast_unset
- smi_isis.h, 36
- smi_isis_adjacency_check_ipv4_set
- smi_isis.h, 36
- smi_isis_adjacency_check_ipv4_unset
- smi_isis.h, 37
- smi_isis_adjacency_check_ipv6_set
- smi_isis.h, 37
- smi_isis_adjacency_check_ipv6_unset
- smi_isis.h, 37
- smi_isis_api_show_ipv6_protocols
- smi_isis.h, 38

- smi_isis_area_password_set
smi_isis.h, [38](#)
- smi_isis_area_password_unset
smi_isis.h, [39](#)
- smi_isis_auth_key_chain_set
smi_isis.h, [39](#)
- smi_isis_auth_key_chain_unset
smi_isis.h, [40](#)
- smi_isis_auth_mode_hmac_md5_set
smi_isis.h, [40](#)
- smi_isis_auth_mode_hmac_md5_unset
smi_isis.h, [41](#)
- smi_isis_auth_mode_text_set
smi_isis.h, [41](#)
- smi_isis_auth_mode_text_unset
smi_isis.h, [42](#)
- smi_isis_auth_send_only_set
smi_isis.h, [42](#)
- smi_isis_auth_send_only_unset
smi_isis.h, [43](#)
- smi_isis_clear_counters
smi_isis.h, [43](#)
- smi_isis_clear_interface_counters
smi_isis.h, [43](#)
- smi_isis_clear_ip_route
smi_isis.h, [44](#)
- smi_isis_clear_ipv6_route
smi_isis.h, [44](#)
- smi_isis_cspf_set
smi_isis.h, [45](#)
- smi_isis_cspf_unset
smi_isis.h, [45](#)
- smi_isis_debug
smi_isis.h, [45](#)
- smi_isis_default_information_originate_
 ipv4_set
smi_isis.h, [46](#)
- smi_isis_default_information_originate_
 ipv4_unset
smi_isis.h, [47](#)
- smi_isis_default_information_originate_
 ipv6_set
smi_isis.h, [47](#)
- smi_isis_default_information_originate_
 ipv6_unset
smi_isis.h, [47](#)
- smi_isis_distance_ipv6_set
smi_isis.h, [48](#)
- smi_isis_distance_ipv6_unset
smi_isis.h, [48](#)
- smi_isis_distance_set
smi_isis.h, [49](#)
- smi_isis_distance_source_set
smi_isis.h, [49](#)
- smi_isis_distance_source_unset
smi_isis.h, [50](#)
- smi_isis_distance_key_unset
smi_isis.h, [50](#)
- smi_isis_domain_password_set
smi_isis.h, [51](#)
- smi_isis_domain_password_unset
smi_isis.h, [51](#)
- smi_isis_get_circ_3way_enabled
smi_isis.h, [51](#)
- smi_isis_get_circ_adj_changes
smi_isis.h, [52](#)
- smi_isis_get_circ_admin_state
smi_isis.h, [52](#)
- smi_isis_get_circ_auth_fails
smi_isis.h, [53](#)
- smi_isis_get_circ_auth_type_fails
smi_isis.h, [53](#)
- smi_isis_get_circ_exist_state
smi_isis.h, [54](#)
- smi_isis_get_circ_ext_domain
smi_isis.h, [54](#)
- smi_isis_get_circ_id_len_mismatches
smi_isis.h, [55](#)
- smi_isis_get_circ_ifindex
smi_isis.h, [55](#)
- smi_isis_get_circ_init_fails
smi_isis.h, [56](#)
- smi_isis_get_circ_lan_dis_changes
smi_isis.h, [56](#)
- smi_isis_get_circ_level
smi_isis.h, [57](#)
- smi_isis_get_circ_level_csnp_interval
smi_isis.h, [57](#)
- smi_isis_get_circ_level_dis
smi_isis.h, [58](#)
- smi_isis_get_circ_level_dis_hello_timer
smi_isis.h, [58](#)
- smi_isis_get_circ_level_hello_multiplier
smi_isis.h, [59](#)
- smi_isis_get_circ_level_hello_timer
smi_isis.h, [59](#)
- smi_isis_get_circ_level_id
smi_isis.h, [60](#)
- smi_isis_get_circ_level_id_octet
smi_isis.h, [60](#)

- smi_isis_get_circ_level_lsp_throttle
smi_isis.h, 61
- smi_isis_get_circ_level_metric
smi_isis.h, 61
- smi_isis_get_circ_level_min_lsp_retrans
smi_isis.h, 62
- smi_isis_get_circ_level_priority
smi_isis.h, 62
- smi_isis_get_circ_level_psnp_interval
smi_isis.h, 63
- smi_isis_get_circ_level_wide_metric
smi_isis.h, 63
- smi_isis_get_circ_max_area_addr -
mismatches
smi_isis.h, 64
- smi_isis_get_circ_mesh_enabled
smi_isis.h, 64
- smi_isis_get_circ_mesh_group
smi_isis.h, 65
- smi_isis_get_circ_num_adj
smi_isis.h, 65
- smi_isis_get_circ_passive_if
smi_isis.h, 66
- smi_isis_get_circ_rej_adj
smi_isis.h, 66
- smi_isis_get_circ_small_hellos
smi_isis.h, 67
- smi_isis_get_circ_type
smi_isis.h, 67
- smi_isis_get_circ_uptime
smi_isis.h, 68
- smi_isis_get_ip_ra_admin_state
smi_isis.h, 68
- smi_isis_get_ip_ra_exist_state
smi_isis.h, 69
- smi_isis_get_ip_ra_full_metric
smi_isis.h, 69
- smi_isis_get_ip_ra_metric
smi_isis.h, 70
- smi_isis_get_ip_ra_metric_type
smi_isis.h, 71
- smi_isis_get_ip_ra_snpa_address
smi_isis.h, 71
- smi_isis_get_ip_ra_source_type
smi_isis.h, 72
- smi_isis_get_ip_ra_type
smi_isis.h, 72
- smi_isis_get_is_adj_3way_state
smi_isis.h, 73
- smi_isis_get_is_adj_area_address
smi_isis.h, 73
- smi_isis_get_is_adj_extended_circ_id
smi_isis.h, 74
- smi_isis_get_is_adj_hold_time
smi_isis.h, 74
- smi_isis_get_is_adj_ip_addr_type
smi_isis.h, 75
- smi_isis_get_is_adj_ip_address
smi_isis.h, 75
- smi_isis_get_is_adj_nbr_priority
smi_isis.h, 76
- smi_isis_get_is_adj_nbr_snpa_addr
smi_isis.h, 76
- smi_isis_get_is_adj_nbr_sys_id
smi_isis.h, 77
- smi_isis_get_is_adj_nbr_sys_type
smi_isis.h, 77
- smi_isis_get_is_adj_prot_supp_protocol
smi_isis.h, 78
- smi_isis_get_is_adj_state
smi_isis.h, 78
- smi_isis_get_is_adj_uptime
smi_isis.h, 79
- smi_isis_get_is_adj_usage
smi_isis.h, 79
- smi_isis_get_lsp_attributes
smi_isis.h, 80
- smi_isis_get_lsp_checksum
smi_isis.h, 80
- smi_isis_get_lsp_lifetime_remain
smi_isis.h, 81
- smi_isis_get_lsp_pdu_length
smi_isis.h, 81
- smi_isis_get_lsp_seq
smi_isis.h, 82
- smi_isis_get_lsp_tlv_checksum
smi_isis.h, 82
- smi_isis_get_lsp_tlv_index
smi_isis.h, 83
- smi_isis_get_lsp_tlv_len
smi_isis.h, 83
- smi_isis_get_lsp_tlv_seq
smi_isis.h, 84
- smi_isis_get_lsp_tlv_type
smi_isis.h, 84
- smi_isis_get_lsp_zero_life
smi_isis.h, 85
- smi_isis_get_man_area_addr_state
smi_isis.h, 85
- smi_isis_get_packet_count_csnp

- smi_isis.h, [86](#)
- smi_isis_get_packet_count_hello
 - smi_isis.h, [87](#)
- smi_isis_get_packet_count_lsp
 - smi_isis.h, [87](#)
- smi_isis_get_packet_count_psnp
 - smi_isis.h, [88](#)
- smi_isis_get_packet_count_unknown
 - smi_isis.h, [88](#)
- smi_isis_get_prot_supp_exist_state
 - smi_isis.h, [89](#)
- smi_isis_get_summ_addr_full_metric
 - smi_isis.h, [89](#)
- smi_isis_get_summ_addr_metric
 - smi_isis.h, [90](#)
- smi_isis_get_summ_addr_state
 - smi_isis.h, [90](#)
- smi_isis_get_sys_admin_state
 - smi_isis.h, [91](#)
- smi_isis_get_sys_area_addr
 - smi_isis.h, [91](#)
- smi_isis_get_sys_exist_state
 - smi_isis.h, [92](#)
- smi_isis_get_sys_id
 - smi_isis.h, [92](#)
- smi_isis_get_sys_l2_to_l1_leaking
 - smi_isis.h, [93](#)
- smi_isis_get_sys_level_lsp_bufsize
 - smi_isis.h, [93](#)
- smi_isis_get_sys_level_metric_style
 - smi_isis.h, [94](#)
- smi_isis_get_sys_level_min_lsp_gen_
interval
 - smi_isis.h, [94](#)
- smi_isis_get_sys_level_overload_state
 - smi_isis.h, [95](#)
- smi_isis_get_sys_level_set_overload
 - smi_isis.h, [95](#)
- smi_isis_get_sys_level_set_overload_
until
 - smi_isis.h, [96](#)
- smi_isis_get_sys_level_spf_considers
 - smi_isis.h, [96](#)
- smi_isis_get_sys_level_te_enabled
 - smi_isis.h, [96](#)
- smi_isis_get_sys_log_adj_changes
 - smi_isis.h, [97](#)
- smi_isis_get_sys_max_age
 - smi_isis.h, [97](#)
- smi_isis_get_sys_max_area_addrs
 - smi_isis.h, [98](#)
- smi_isis_get_sys_max_lsp_gen_interval
 - smi_isis.h, [98](#)
- smi_isis_get_sys_max_path_splits
 - smi_isis.h, [99](#)
- smi_isis_get_sys_next_circ_index
 - smi_isis.h, [99](#)
- smi_isis_get_sys_poll_es_hello_rate
 - smi_isis.h, [99](#)
- smi_isis_get_sys_receive_lsp_bufsize
 - smi_isis.h, [100](#)
- smi_isis_get_sys_stat_auth_fails
 - smi_isis.h, [100](#)
- smi_isis_get_sys_stat_auth_type_fails
 - smi_isis.h, [101](#)
- smi_isis_get_sys_stat_corrupted_lsps
 - smi_isis.h, [101](#)
- smi_isis_get_sys_stat_exceed_max_
seqnums
 - smi_isis.h, [102](#)
- smi_isis_get_sys_stat_id_len_
mismatches
 - smi_isis.h, [102](#)
- smi_isis_get_sys_stat_lsp_purges
 - smi_isis.h, [103](#)
- smi_isis_get_sys_stat_lspdb_overloaded
 - smi_isis.h, [103](#)
- smi_isis_get_sys_stat_man_addr_drop_
area
 - smi_isis.h, [104](#)
- smi_isis_get_sys_stat_max_area_addr_
mismatches
 - smi_isis.h, [104](#)
- smi_isis_get_sys_stat_partition_changes
 - smi_isis.h, [105](#)
- smi_isis_get_sys_stat_seqnum_skips
 - smi_isis.h, [105](#)
- smi_isis_get_sys_stat_spf_runs
 - smi_isis.h, [106](#)
- smi_isis_get_sys_type
 - smi_isis.h, [106](#)
- smi_isis_get_sys_version
 - smi_isis.h, [107](#)
- smi_isis_get_sys_wait_time
 - smi_isis.h, [107](#)
- smi_isis_high_priority_tag_set
 - smi_isis.h, [107](#)
- smi_isis_high_priority_tag_unset
 - smi_isis.h, [108](#)
- smi_isis_hostname_dynamic_set

- smi_isis.h, [108](#)
- smi_isis_hostname_dynamic_unset
 - smi_isis.h, [109](#)
- smi_isis_if_auth_key_chain_set
 - smi_isis.h, [109](#)
- smi_isis_if_auth_key_chain_unset
 - smi_isis.h, [110](#)
- smi_isis_if_auth_mode_hmac_md5_set
 - smi_isis.h, [110](#)
- smi_isis_if_auth_mode_hmac_md5_unset
 - smi_isis.h, [111](#)
- smi_isis_if_auth_mode_text_set
 - smi_isis.h, [111](#)
- smi_isis_if_auth_mode_text_unset
 - smi_isis.h, [112](#)
- smi_isis_if_auth_send_only_set
 - smi_isis.h, [112](#)
- smi_isis_if_auth_send_only_unset
 - smi_isis.h, [113](#)
- smi_isis_if_circuit_type_set
 - smi_isis.h, [113](#)
- smi_isis_if_circuit_type_unset
 - smi_isis.h, [114](#)
- smi_isis_if_csnp_interval_set
 - smi_isis.h, [114](#)
- smi_isis_if_csnp_interval_unset
 - smi_isis.h, [114](#)
- smi_isis_if_hello_interval_minimal_set
 - smi_isis.h, [115](#)
- smi_isis_if_hello_interval_set
 - smi_isis.h, [115](#)
- smi_isis_if_hello_interval_unset
 - smi_isis.h, [116](#)
- smi_isis_if_hello_multiplier_set
 - smi_isis.h, [116](#)
- smi_isis_if_hello_multiplier_unset
 - smi_isis.h, [117](#)
- smi_isis_if_hello_padding_set
 - smi_isis.h, [117](#)
- smi_isis_if_hello_padding_unset
 - smi_isis.h, [118](#)
- smi_isis_if_ip_router_set
 - smi_isis.h, [118](#)
- smi_isis_if_ip_router_unset
 - smi_isis.h, [119](#)
- smi_isis_if_ipv6_router_set
 - smi_isis.h, [119](#)
- smi_isis_if_ipv6_router_unset
 - smi_isis.h, [119](#)
- smi_isis_if_lsp_interval_set
 - smi_isis.h, [120](#)
- smi_isis_if_lsp_interval_unset
 - smi_isis.h, [120](#)
- smi_isis_if_mesh_group_block_set
 - smi_isis.h, [121](#)
- smi_isis_if_mesh_group_set
 - smi_isis.h, [121](#)
- smi_isis_if_mesh_group_unset
 - smi_isis.h, [121](#)
- smi_isis_if_metric_set
 - smi_isis.h, [122](#)
- smi_isis_if_metric_unset
 - smi_isis.h, [122](#)
- smi_isis_if_network_type_set
 - smi_isis.h, [123](#)
- smi_isis_if_network_type_unset
 - smi_isis.h, [123](#)
- smi_isis_if_password_set
 - smi_isis.h, [124](#)
- smi_isis_if_password_unset
 - smi_isis.h, [124](#)
- smi_isis_if_priority_set
 - smi_isis.h, [125](#)
- smi_isis_if_priority_unset
 - smi_isis.h, [125](#)
- smi_isis_if_retransmit_interval_set
 - smi_isis.h, [126](#)
- smi_isis_if_tag_set
 - smi_isis.h, [126](#)
- smi_isis_if_tag_unset
 - smi_isis.h, [127](#)
- smi_isis_if_wide_metric_set
 - smi_isis.h, [127](#)
- smi_isis_if_wide_metric_unset
 - smi_isis.h, [128](#)
- smi_isis_ignore_lsp_errors_set
 - smi_isis.h, [128](#)
- smi_isis_ignore_lsp_errors_unset
 - smi_isis.h, [129](#)
- smi_isis_instance_set
 - smi_isis.h, [129](#)
- smi_isis_instance_unset
 - smi_isis.h, [129](#)
- smi_isis_instance_unset_restart
 - smi_isis.h, [130](#)
- smi_isis_is_type_set
 - smi_isis.h, [130](#)
- smi_isis_is_type_unset
 - smi_isis.h, [131](#)

- smi_isis_ispf_set
 - smi_isis.h, [131](#)
- smi_isis_ispf_unset
 - smi_isis.h, [132](#)
- smi_isis_l1_snp_auth_send_only
 - smi_isis.h, [132](#)
- smi_isis_l1_snp_auth_validate_set
 - smi_isis.h, [132](#)
- smi_isis_l2_snp_auth_send_only
 - smi_isis.h, [133](#)
- smi_isis_l2_snp_auth_validate_set
 - smi_isis.h, [133](#)
- smi_isis_lsp_gen_interval_set
 - smi_isis.h, [134](#)
- smi_isis_lsp_gen_interval_unset
 - smi_isis.h, [134](#)
- smi_isis_lsp_mtu_set
 - smi_isis.h, [135](#)
- smi_isis_lsp_mtu_unset
 - smi_isis.h, [135](#)
- smi_isis_lsp_refresh_interval_set
 - smi_isis.h, [136](#)
- smi_isis_lsp_refresh_interval_unset
 - smi_isis.h, [136](#)
- smi_isis_max_area_addr_set
 - smi_isis.h, [137](#)
- smi_isis_max_area_addr_unset
 - smi_isis.h, [137](#)
- smi_isis_max_lsp_lifetime_set
 - smi_isis.h, [137](#)
- smi_isis_max_lsp_lifetime_unset
 - smi_isis.h, [138](#)
- smi_isis_metric_style_set
 - smi_isis.h, [138](#)
- smi_isis_metric_style_transition_-
 - narrow_set
 - smi_isis.h, [139](#)
- smi_isis_metric_style_transition_set
 - smi_isis.h, [139](#)
- smi_isis_metric_style_transition_wide_-
 - set
 - smi_isis.h, [140](#)
- smi_isis_metric_style_unset
 - smi_isis.h, [140](#)
- smi_isis_mpls_traffic_eng_router_id_set
 - smi_isis.h, [141](#)
- smi_isis_mpls_traffic_eng_router_id_-
 - unset
 - smi_isis.h, [141](#)
- smi_isis_mpls_traffic_eng_set
 - smi_isis.h, [142](#)
- smi_isis_mpls_traffic_eng_unset
 - smi_isis.h, [142](#)
- smi_isis_multi_topology_set
 - smi_isis.h, [143](#)
- smi_isis_multi_topology_transition_set
 - smi_isis.h, [143](#)
- smi_isis_multi_topology_unset
 - smi_isis.h, [144](#)
- smi_isis_net_set
 - smi_isis.h, [144](#)
- smi_isis_net_unset
 - smi_isis.h, [145](#)
- smi_isis_no_debug
 - smi_isis.h, [145](#)
- smi_isis_parse_sys_id
 - smi_isis.h, [146](#)
- smi_isis_passive_interface_default_set
 - smi_isis.h, [146](#)
- smi_isis_passive_interface_default_unset
 - smi_isis.h, [147](#)
- smi_isis_passive_interface_set
 - smi_isis.h, [147](#)
- smi_isis_passive_interface_unset
 - smi_isis.h, [148](#)
- smi_isis_prc_interval_set
 - smi_isis.h, [148](#)
- smi_isis_proc_clear
 - smi_isis.h, [148](#)
- smi_isis_protocol_topology_set
 - smi_isis.h, [149](#)
- smi_isis_protocol_topology_unset
 - smi_isis.h, [149](#)
- smi_isis_redistribute_inter_level_ipv4_-
 - set
 - smi_isis.h, [150](#)
- smi_isis_redistribute_inter_level_ipv4_-
 - unset
 - smi_isis.h, [150](#)
- smi_isis_redistribute_inter_level_ipv6_-
 - set
 - smi_isis.h, [151](#)
- smi_isis_redistribute_inter_level_ipv6_-
 - unset
 - smi_isis.h, [151](#)
- smi_isis_redistribute_ipv4_set
 - smi_isis.h, [152](#)
- smi_isis_redistribute_ipv4_unset
 - smi_isis.h, [152](#)
- smi_isis_redistribute_ipv6_set

- smi_isis.h, [153](#)
- smi_isis_redistribute_ipv6_unset
 - smi_isis.h, [153](#)
- smi_isis_restart_grace_period_set
 - smi_isis.h, [154](#)
- smi_isis_restart_grace_period_unset
 - smi_isis.h, [154](#)
- smi_isis_restart_hello_interval_set
 - smi_isis.h, [155](#)
- smi_isis_restart_hello_interval_unset
 - smi_isis.h, [155](#)
- smi_isis_restart_helper_set
 - smi_isis.h, [155](#)
- smi_isis_restart_helper_unset
 - smi_isis.h, [156](#)
- smi_isis_restart_level_timer_set
 - smi_isis.h, [156](#)
- smi_isis_restart_level_timer_unset
 - smi_isis.h, [157](#)
- smi_isis_restart_set
 - smi_isis.h, [157](#)
- smi_isis_restart_suppress_adjacency_set
 - smi_isis.h, [157](#)
- smi_isis_restart_suppress_adjacency_-
 - unset
 - smi_isis.h, [158](#)
- smi_isis_set_circ_3way_enabled
 - smi_isis.h, [158](#)
- smi_isis_set_circ_admin_state
 - smi_isis.h, [159](#)
- smi_isis_set_circ_exist_state
 - smi_isis.h, [159](#)
- smi_isis_set_circ_ext_domain
 - smi_isis.h, [159](#)
- smi_isis_set_circ_ifindex
 - smi_isis.h, [160](#)
- smi_isis_set_circ_level
 - smi_isis.h, [160](#)
- smi_isis_set_circ_level_dis_hello_timer
 - smi_isis.h, [161](#)
- smi_isis_set_circ_level_hello_multiplier
 - smi_isis.h, [161](#)
- smi_isis_set_circ_level_hello_timer
 - smi_isis.h, [162](#)
- smi_isis_set_circ_level_id_octet
 - smi_isis.h, [163](#)
- smi_isis_set_circ_level_lsp_throttle
 - smi_isis.h, [163](#)
- smi_isis_set_circ_level_metric
 - smi_isis.h, [164](#)
- smi_isis_set_circ_level_wide_metric
 - smi_isis.h, [164](#)
- smi_isis_set_circ_mesh_enabled
 - smi_isis.h, [165](#)
- smi_isis_set_circ_mesh_group
 - smi_isis.h, [165](#)
- smi_isis_set_circ_passive_if
 - smi_isis.h, [166](#)
- smi_isis_set_circ_small_hellos
 - smi_isis.h, [166](#)
- smi_isis_set_circ_type
 - smi_isis.h, [167](#)
- smi_isis_set_ip_ra_admin_state
 - smi_isis.h, [167](#)
- smi_isis_set_ip_ra_exist_state
 - smi_isis.h, [168](#)
- smi_isis_set_ip_ra_full_metric
 - smi_isis.h, [168](#)
- smi_isis_set_ip_ra_metric
 - smi_isis.h, [169](#)
- smi_isis_set_ip_ra_metric_type
 - smi_isis.h, [170](#)
- smi_isis_set_ip_ra_nexthop_type
 - smi_isis.h, [170](#)
- smi_isis_set_ip_ra_type
 - smi_isis.h, [171](#)
- smi_isis_set_man_area_addr_state
 - smi_isis.h, [171](#)
- smi_isis_set_prot_supp_exist_state
 - smi_isis.h, [172](#)
- smi_isis_set_sys_admin_state
 - smi_isis.h, [172](#)
- smi_isis_set_sys_exist_state
 - smi_isis.h, [173](#)
- smi_isis_set_sys_l2_to_l1_leaking
 - smi_isis.h, [173](#)
- smi_isis_set_sys_level_lsp_bufsize
 - smi_isis.h, [174](#)
- smi_isis_set_sys_level_set_overload
 - smi_isis.h, [174](#)
- smi_isis_set_sys_level_set_overload_-
 - until
 - smi_isis.h, [175](#)
- smi_isis_set_sys_level_spf_considers
 - smi_isis.h, [175](#)
- smi_isis_set_sys_level_te_enabled
 - smi_isis.h, [176](#)
- smi_isis_set_sys_log_adj_changes
 - smi_isis.h, [176](#)
- smi_isis_set_sys_max_age

smi_isis.h, [177](#)
smi_isis_set_sys_max_area_addr
smi_isis.h, [177](#)
smi_isis_set_sys_max_lsp_gen_interval
smi_isis.h, [178](#)
smi_isis_set_sys_max_path_splits
smi_isis.h, [178](#)
smi_isis_set_sys_poll_es_hello_rate
smi_isis.h, [178](#)
smi_isis_set_sys_receive_lsp_bufsize
smi_isis.h, [179](#)
smi_isis_set_sys_type
smi_isis.h, [179](#)
smi_isis_set_sys_wait_time
smi_isis.h, [180](#)
smi_isis_show_clsns_if_nbr_api
smi_isis.h, [180](#)
smi_isis_show_clsns_nbr_api
smi_isis.h, [181](#)
smi_isis_show_clsns_neighbors_api
smi_isis.h, [181](#)
smi_isis_show_database
smi_isis.h, [182](#)
smi_isis_show_database_filtered
smi_isis.h, [182](#)
smi_isis_show_global_stat
smi_isis.h, [183](#)
smi_isis_show_if_stat
smi_isis.h, [183](#)
smi_isis_show_tag_global_stat
smi_isis.h, [184](#)
smi_isis_show_tag_if_stat
smi_isis.h, [184](#)
smi_isis_show_topology_all
smi_isis.h, [185](#)
smi_isis_spf_interval_set
smi_isis.h, [185](#)
smi_isis_spf_interval_unset
smi_isis.h, [186](#)
smi_isis_summary_address_set
smi_isis.h, [186](#)
smi_isis_summary_address_unset
smi_isis.h, [187](#)
smi_show_ip_isis_route
smi_isis.h, [187](#)
smi_show_ipv6_isis_route
smi_isis.h, [188](#)
smi_show_isis_interface
smi_isis.h, [188](#)