

ZebOS-XP OSPFv3 SMI Reference
IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:50 2015

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	smi_ospf6.h File Reference	3
2.1.1	Detailed Description	46
2.1.2	Function Documentation	46
2.1.2.1	merge_ospf6_list_nbr_info	46
2.1.2.2	smi_debug_no_ospf6_packet	46
2.1.2.3	smi_debug_ospf6_packet	47
2.1.2.4	smi_ospf6_abr_type_set	48
2.1.2.5	smi_ospf6_abr_type_unset	48
2.1.2.6	smi_ospf6_address_family_set	49
2.1.2.7	smi_ospf6_address_family_unset	49
2.1.2.8	smi_ospf6_area_default_cost_set	50
2.1.2.9	smi_ospf6_area_default_cost_unset	50
2.1.2.10	smi_ospf6_area_format_set	51
2.1.2.11	smi_ospf6_area_no_summary_set	51
2.1.2.12	smi_ospf6_area_nssa_default_originate_metric_set	52
2.1.2.13	smi_ospf6_area_nssa_default_originate_metric_- type_set	52
2.1.2.14	smi_ospf6_area_nssa_default_originate_set	53
2.1.2.15	smi_ospf6_area_nssa_default_originate_unset	53
2.1.2.16	smi_ospf6_area_nssa_no_redistribution_set	54
2.1.2.17	smi_ospf6_area_nssa_no_redistribution_unset	54
2.1.2.18	smi_ospf6_area_nssa_set	55

2.1.2.19	smi_ospf6_area_nssa_stability_interval_set	55
2.1.2.20	smi_ospf6_area_nssa_translator_role_set	56
2.1.2.21	smi_ospf6_area_nssa_translator_role_unset	56
2.1.2.22	smi_ospf6_area_nssa_unset	57
2.1.2.23	smi_ospf6_area_range_ipv4_set_sdkapi	57
2.1.2.24	smi_ospf6_area_range_ipv4_unset_sdkapi	58
2.1.2.25	smi_ospf6_area_stub_no_summary_set	59
2.1.2.26	smi_ospf6_area_stub_set	60
2.1.2.27	smi_ospf6_area_stub_unset	60
2.1.2.28	smi_ospf6_auto_cost_reference_bandwidth_set	60
2.1.2.29	smi_ospf6_auto_cost_reference_bandwidth_unset	61
2.1.2.30	smi_ospf6_capability_cspf_set	61
2.1.2.31	smi_ospf6_capability_cspf_unset	62
2.1.2.32	smi_ospf6_capability_restart_set	62
2.1.2.33	smi_ospf6_capability_restart_unset	63
2.1.2.34	smi_ospf6_debug	63
2.1.2.35	smi_ospf6_default_metric_set	64
2.1.2.36	smi_ospf6_default_metric_unset	65
2.1.2.37	smi_ospf6_disable_db_summary_opt	65
2.1.2.38	smi_ospf6_distance_all_set	66
2.1.2.39	smi_ospf6_distance_all_unset	66
2.1.2.40	smi_ospf6_distance_external_set	66
2.1.2.41	smi_ospf6_distance_external_unset	67
2.1.2.42	smi_ospf6_distance_inter_area_set	67
2.1.2.43	smi_ospf6_distance_inter_area_unset	68
2.1.2.44	smi_ospf6_distance_intra_area_set	68
2.1.2.45	smi_ospf6_distance_intra_area_unset	68
2.1.2.46	smi_ospf6_distribute_list_in_set	69
2.1.2.47	smi_ospf6_distribute_list_in_unset	69
2.1.2.48	smi_ospf6_distribute_list_out_set	70
2.1.2.49	smi_ospf6_distribute_list_out_unset	70
2.1.2.50	smi_ospf6_enable_db_summary_opt	71
2.1.2.51	smi_ospf6_get_admin_stat	71
2.1.2.52	smi_ospf6_get_area_bdr_rtr_count	72

2.1.2.53	smi_ospf6_get_area_bdr_rtr_status	72
2.1.2.54	smi_ospf6_get_area_lsa_cksum_sum	73
2.1.2.55	smi_ospf6_get_area_lsa_count	73
2.1.2.56	smi_ospf6_get_area_lsdb_advertisement	73
2.1.2.57	smi_ospf6_get_area_lsdb_age	74
2.1.2.58	smi_ospf6_get_area_lsdb_checksum	74
2.1.2.59	smi_ospf6_get_area_lsdb_sequence	75
2.1.2.60	smi_ospf6_get_area_lsdb_type_known	75
2.1.2.61	smi_ospf6_get_area_nssa_trans_events	76
2.1.2.62	smi_ospf6_get_area_nssa_trans_role	76
2.1.2.63	smi_ospf6_get_area_nssa_trans_stability_interval	77
2.1.2.64	smi_ospf6_get_area_nssa_trans_state	77
2.1.2.65	smi_ospf6_get_area_status	78
2.1.2.66	smi_ospf6_get_area_stub_metric_type	78
2.1.2.67	smi_ospf6_get_area_summary	78
2.1.2.68	smi_ospf6_get_area_te_enabled	79
2.1.2.69	smi_ospf6_get_as_lsdb_advertisement	79
2.1.2.70	smi_ospf6_get_as_lsdb_age	80
2.1.2.71	smi_ospf6_get_as_lsdb_checksum	80
2.1.2.72	smi_ospf6_get_as_lsdb_sequence	81
2.1.2.73	smi_ospf6_get_as_lsdb_type_known	81
2.1.2.74	smi_ospf6_get_as_scope_lsa_cksumsum	82
2.1.2.75	smi_ospf6_get_as_scope_lsa_count	82
2.1.2.76	smi_ospf6_get_asbdr_rtr_count	82
2.1.2.77	smi_ospf6_get_asbdr_rtr_status	83
2.1.2.78	smi_ospf6_get_discontinuity_time	83
2.1.2.79	smi_ospf6_get_extern_lsa_count	83
2.1.2.80	smi_ospf6_get_if_admin_stat	84
2.1.2.81	smi_ospf6_get_if_area_id	84
2.1.2.82	smi_ospf6_get_if_bdr	85
2.1.2.83	smi_ospf6_get_if_demand	85
2.1.2.84	smi_ospf6_get_if_demand_nbr_probe	86
2.1.2.85	smi_ospf6_get_if_demand_nbr_probe_interval	86
2.1.2.86	smi_ospf6_get_if_demand_nbr_probe_retrans_limit	86

2.1.2.87	smi_ospf6_get_if_dr	87
2.1.2.88	smi_ospf6_get_if_events	87
2.1.2.89	smi_ospf6_get_if_hello_interval	88
2.1.2.90	smi_ospf6_get_if_link_lsa_cksumsum	88
2.1.2.91	smi_ospf6_get_if_link_lsa_suppression	89
2.1.2.92	smi_ospf6_get_if_link_scope_lsa_count	89
2.1.2.93	smi_ospf6_get_if_metric_value	89
2.1.2.94	smi_ospf6_get_if_poll_interval	90
2.1.2.95	smi_ospf6_get_if_retrans_interval	90
2.1.2.96	smi_ospf6_get_if_rtr_dead_interval	91
2.1.2.97	smi_ospf6_get_if_rtr_priority	91
2.1.2.98	smi_ospf6_get_if_state	92
2.1.2.99	smi_ospf6_get_if_status	92
2.1.2.100	smi_ospf6_get_if_te_disabled	92
2.1.2.101	smi_ospf6_get_if_transit_delay	93
2.1.2.102	smi_ospf6_get_if_type	93
2.1.2.103	smi_ospf6_get_import_as_extern	94
2.1.2.104	smi_ospf6_get_link_lsdb_advertisement	94
2.1.2.105	smi_ospf6_get_link_lsdb_age	95
2.1.2.106	smi_ospf6_get_link_lsdb_checksum	95
2.1.2.107	smi_ospf6_get_link_lsdb_sequence	96
2.1.2.108	smi_ospf6_get_link_lsdb_type_known	96
2.1.2.109	smi_ospf6_get_nbr_address	97
2.1.2.110	smi_ospf6_get_nbr_address_type	97
2.1.2.111	smi_ospf6_get_nbr_events	98
2.1.2.112	smi_ospf6_get_nbr_hello_suppressed	98
2.1.2.113	smi_ospf6_get_nbr_if_id	99
2.1.2.114	smi_ospf6_get_nbr_lsretransq_len	99
2.1.2.115	smi_ospf6_get_nbr_options	100
2.1.2.116	smi_ospf6_get_nbr_priority	100
2.1.2.117	smi_ospf6_get_nbr_restart_helper_age	101
2.1.2.118	smi_ospf6_get_nbr_restart_helper_exit_reason . . .	101
2.1.2.119	smi_ospf6_get_nbr_restart_helper_status	102
2.1.2.120	smi_ospf6_get_nbr_state	102

2.1.2.121 smi_ospf6_get_notification_enable	103
2.1.2.122 smi_ospf6_get_orinate_new_lsas	103
2.1.2.123 smi_ospf6_get_reference_bandwidth	103
2.1.2.124 smi_ospf6_get_restart_age	104
2.1.2.125 smi_ospf6_get_restart_exit_reason	104
2.1.2.126 smi_ospf6_get_restart_interval	104
2.1.2.127 smi_ospf6_get_restart_status	105
2.1.2.128 smi_ospf6_get_restart_strict_lsa_check	105
2.1.2.129 smi_ospf6_get_restart_support	106
2.1.2.130 smi_ospf6_get_restart_time	106
2.1.2.131 smi_ospf6_get_router_id	106
2.1.2.132 smi_ospf6_get_rx_new_lsas	107
2.1.2.133 smi_ospf6_get_spf_runs	107
2.1.2.134 smi_ospf6_get_stub_metric	107
2.1.2.135 smi_ospf6_get_stub_router_advertisement	108
2.1.2.136 smi_ospf6_get_stub_router_support	108
2.1.2.137 smi_ospf6_get_version_num	108
2.1.2.138 smi_ospf6_get_virt_if_events	109
2.1.2.139 smi_ospf6_get_virt_if_hello_interval	109
2.1.2.140 smi_ospf6_get_virt_if_index	110
2.1.2.141 smi_ospf6_get_virt_if_instid	110
2.1.2.142 smi_ospf6_get_virt_if_link_lsa_cksumsum	110
2.1.2.143 smi_ospf6_get_virt_if_link_scope_lsa_count	111
2.1.2.144 smi_ospf6_get_virt_if_retrans_interval	111
2.1.2.145 smi_ospf6_get_virt_if_rtr_dead_interval	112
2.1.2.146 smi_ospf6_get_virt_if_state	112
2.1.2.147 smi_ospf6_get_virt_if_status	113
2.1.2.148 smi_ospf6_get_virt_if_transmit_delay	113
2.1.2.149 smi_ospf6_get_virt_link_lsdb_advertisement	114
2.1.2.150 smi_ospf6_get_virt_link_lsdb_age	114
2.1.2.151 smi_ospf6_get_virt_link_lsdb_checksum	115
2.1.2.152 smi_ospf6_get_virt_link_lsdb_sequence	115
2.1.2.153 smi_ospf6_get_virt_link_lsdb_type_known	116
2.1.2.154 smi_ospf6_get_virt_nbr_address	117

2.1.2.155 smi_ospf6_get_virt_nbr_address_type	117
2.1.2.156 smi_ospf6_get_virt_nbr_events	117
2.1.2.157 smi_ospf6_get_virt_nbr_hello_suppressed	118
2.1.2.158 smi_ospf6_get_virt_nbr_if_id	118
2.1.2.159 smi_ospf6_get_virt_nbr_ifindex	119
2.1.2.160 smi_ospf6_get_virt_nbr_ifinstid	120
2.1.2.161 smi_ospf6_get_virt_nbr_ls_retransq_len	120
2.1.2.162 smi_ospf6_get_virt_nbr_options	120
2.1.2.163 smi_ospf6_get_virt_nbr_restart_helper_age	121
2.1.2.164 smi_ospf6_get_virt_nbr_restart_helper_exit_reason	121
2.1.2.165 smi_ospf6_get_virt_nbr_restart_helper_status	122
2.1.2.166 smi_ospf6_get_virt_nbr_state	122
2.1.2.167 smi_ospf6_graceful_restart_set	123
2.1.2.168 smi_ospf6_graceful_restart_unset	123
2.1.2.169 smi_ospf6_if_cost_set	124
2.1.2.170 smi_ospf6_if_cost_unset	124
2.1.2.171 smi_ospf6_if_dead_interval_set	124
2.1.2.172 smi_ospf6_if_dead_interval_unset	125
2.1.2.173 smi_ospf6_if_disable_all_set	125
2.1.2.174 smi_ospf6_if_disable_all_unset	126
2.1.2.175 smi_ospf6_if_hello_interval_set	126
2.1.2.176 smi_ospf6_if_hello_interval_unset	126
2.1.2.177 smi_ospf6_if_ipv6_router_set	127
2.1.2.178 smi_ospf6_if_ipv6_router_unset	127
2.1.2.179 smi_ospf6_if_link_lsa_suppression_set	128
2.1.2.180 smi_ospf6_if_mtu_ignore_set	129
2.1.2.181 smi_ospf6_if_mtu_ignore_unset	129
2.1.2.182 smi_ospf6_if_network_set	130
2.1.2.183 smi_ospf6_if_network_unset	130
2.1.2.184 smi_ospf6_if_priority_set	131
2.1.2.185 smi_ospf6_if_priority_unset	131
2.1.2.186 smi_ospf6_if_retransmit_interval_set	131
2.1.2.187 smi_ospf6_if_retransmit_interval_unset	132
2.1.2.188 smi_ospf6_if_te_metric_set	132

2.1.2.189 smi_ospf6_if_te_metric_unset	133
2.1.2.190 smi_ospf6_if_transmit_delay_set	133
2.1.2.191 smi_ospf6_if_transmit_delay_unset	134
2.1.2.192 smi_ospf6_ipv6_ospf_display_route_single_line_set	134
2.1.2.193 smi_ospf6_ipv6_ospf_display_route_single_line_- unset	134
2.1.2.194 smi_ospf6_log_adjacency_changes_set	135
2.1.2.195 smi_ospf6_log_adjacency_changes_unset	135
2.1.2.196 smi_ospf6_max_concurrent_dd_set	136
2.1.2.197 smi_ospf6_max_concurrent_dd_unset	136
2.1.2.198 smi_ospf6_max_unuse_lsa_set	137
2.1.2.199 smi_ospf6_max_unuse_lsa_unset	137
2.1.2.200 smi_ospf6_max_unuse_packet_set	137
2.1.2.201 smi_ospf6_max_unuse_packet_unset	138
2.1.2.202 smi_ospf6_no_debug	138
2.1.2.203 smi_ospf6_passive_if_default_set	140
2.1.2.204 smi_ospf6_passive_if_default_unset	140
2.1.2.205 smi_ospf6_passive_if_set	140
2.1.2.206 smi_ospf6_passive_if_unset	141
2.1.2.207 smi_ospf6_process_shutdown_set	141
2.1.2.208 smi_ospf6_process_shutdown_unset	142
2.1.2.209 smi_ospf6_redis_default_set	142
2.1.2.210 smi_ospf6_redis_default_unset	142
2.1.2.211 smi_ospf6_redistribute_metric_set_by_ospf6_src_- tag_sdkapi	143
2.1.2.212 smi_ospf6_redistribute_metric_set_by_ospf_src_- tag_sdkapi	143
2.1.2.213 smi_ospf6_redistribute_metric_type_set_by_- ospf6_src_tag_sdkapi	144
2.1.2.214 smi_ospf6_redistribute_metric_type_set_by_ospf_- src_tag_sdkapi	145
2.1.2.215 smi_ospf6_redistribute_metric_type_unset_by_- ospf6_src_tag_sdkapi	146
2.1.2.216 smi_ospf6_redistribute_metric_type_unset_by_- ospf_src_tag_sdkapi	146

2.1.2.217 smi_ospf6_redistribute_metric_unset_by_ospf6_- src_tag_sdkapi	147
2.1.2.218 smi_ospf6_redistribute_metric_unset_by_ospf_- src_tag_sdkapi	147
2.1.2.219 smi_ospf6_redistribute_route_tag_set_by_ospf6_- src_tag_sdkapi	148
2.1.2.220 smi_ospf6_redistribute_route_tag_set_by_ospf_- src_tag_sdkapi	149
2.1.2.221 smi_ospf6_redistribute_route_tag_unset_by_- ospf6_src_tag_sdkapi	149
2.1.2.222 smi_ospf6_redistribute_route_tag_unset_by_ospf_- src_tag_sdkapi	150
2.1.2.223 smi_ospf6_redistribute_set_by_ospf6_src_tag_sdkapi	151
2.1.2.224 smi_ospf6_redistribute_set_by_ospf_src_tag_sdkapi	151
2.1.2.225 smi_ospf6_redistribute_unset_by_ospf6_src_tag_- sdkapi	152
2.1.2.226 smi_ospf6_redistribute_unset_by_ospf_src_tag_- sdkapi	153
2.1.2.227 smi_ospf6_restart_graceful_set	153
2.1.2.228 smi_ospf6_restart_helper_grace_period_set	154
2.1.2.229 smi_ospf6_restart_helper_grace_period_unset . . .	154
2.1.2.230 smi_ospf6_restart_helper_never_router_id_set . . .	154
2.1.2.231 smi_ospf6_restart_helper_never_router_id_unset . .	155
2.1.2.232 smi_ospf6_restart_helper_never_router_unset_all .	155
2.1.2.233 smi_ospf6_restart_helper_policy_set	156
2.1.2.234 smi_ospf6_restart_helper_policy_unset	156
2.1.2.235 smi_ospf6_restart_helper_policy_unset_all	156
2.1.2.236 smi_ospf6_routemap_default_set	157
2.1.2.237 smi_ospf6_routemap_default_unset	157
2.1.2.238 smi_ospf6_routemap_set_by_ospf6_src_tag_sdkapi	158
2.1.2.239 smi_ospf6_routemap_set_by_ospf_src_tag_sdkapi .	158
2.1.2.240 smi_ospf6_routemap_unset_by_ospf6_src_tag_sdkapi	159
2.1.2.241 smi_ospf6_routemap_unset_by_ospf_src_tag_sdkapi	159
2.1.2.242 smi_ospf6_router_id_set	160
2.1.2.243 smi_ospf6_router_id_unset	160
2.1.2.244 smi_ospf6_router_set	161

2.1.2.245 smi_ospf6_router_unset	161
2.1.2.246 smi_ospf6_set_area_nssa_trans_role	162
2.1.2.247 smi_ospf6_set_area_stub_metric_type	162
2.1.2.248 smi_ospf6_set_area_te_enabled	163
2.1.2.249 smi_ospf6_set_asbdr_rtr_status	163
2.1.2.250 smi_ospf6_set_if_admin_stat	164
2.1.2.251 smi_ospf6_set_if_rtr_dead_interval	164
2.1.2.252 smi_ospf6_set_if_type	165
2.1.2.253 smi_ospf6_show_border_routers	165
2.1.2.254 smi_ospf6_show_database	166
2.1.2.255 smi_ospf6_show_debug	166
2.1.2.256 smi_ospf6_show_interface	167
2.1.2.257 smi_ospf6_show_interface_brief	167
2.1.2.258 smi_ospf6_show_interface_brief_by_tag	168
2.1.2.259 smi_ospf6_show_interface_by_tag	168
2.1.2.260 smi_ospf6_show_neighbor_by_if_nbr_id	169
2.1.2.261 smi_ospf6_show_neighbor_by_interface_name	169
2.1.2.262 smi_ospf6_show_neighbor_by_nbr_id	170
2.1.2.263 smi_ospf6_show_neighbor_detail_by_if_nbr_id	170
2.1.2.264 smi_ospf6_show_neighbor_detail_by_interface_name	171
2.1.2.265 smi_ospf6_show_neighbor_detail_by_tag	171
2.1.2.266 smi_ospf6_show_neighbor_summary_by_if_nbr_id	172
2.1.2.267 smi_ospf6_show_neighbor_summary_by_- interface_name	172
2.1.2.268 smi_ospf6_show_neighbor_summary_by_nbr_id	173
2.1.2.269 smi_ospf6_show_neighbor_summary_by_tag	173
2.1.2.270 smi_ospf6_show_process	174
2.1.2.271 smi_ospf6_show_route	174
2.1.2.272 smi_ospf6_show_route_by_prefix	175
2.1.2.273 smi_ospf6_show_route_summary	175
2.1.2.274 smi_ospf6_show_virtual_link	176
2.1.2.275 smi_ospf6_show_virtual_link_brief	176
2.1.2.276 smi_ospf6_summary_address_ipv4_set_sdkapi	177
2.1.2.277 smi_ospf6_summary_address_ipv4_unset_sdkapi	177

2.1.2.278	smi_ospf6_summary_address_not_advertise_- ipv4_set_sdkapi	178
2.1.2.279	smi_ospf6_summary_address_not_advertise_- ipv4_unset_sdkapi	179
2.1.2.280	smi_ospf6_summary_address_not_advertise_- ipv6_unset_sdkapi	180
2.1.2.281	smi_ospf6_summary_address_tag_ipv4_set_sdkapi	180
2.1.2.282	smi_ospf6_summary_address_tag_ipv4_unset_sdkapi	181
2.1.2.283	smi_ospf6_timers_spf_set	182
2.1.2.284	smi_ospf6_timers_spf_unset	182
2.1.2.285	smi_ospf6_vlink_dead_interval_set	183
2.1.2.286	smi_ospf6_vlink_dead_interval_unset	183
2.1.2.287	smi_ospf6_vlink_hello_interval_set	184
2.1.2.288	smi_ospf6_vlink_hello_interval_unset	184
2.1.2.289	smi_ospf6_vlink_instance_id_set	185
2.1.2.290	smi_ospf6_vlink_instance_id_unset	185
2.1.2.291	smi_ospf6_vlink_retransmit_interval_set	186
2.1.2.292	smi_ospf6_vlink_retransmit_interval_unset	186
2.1.2.293	smi_ospf6_vlink_set	187
2.1.2.294	smi_ospf6_vlink_transmit_delay_set	187
2.1.2.295	smi_ospf6_vlink_transmit_delay_unset	188
2.1.2.296	smi_ospf6_vlink_unset	188
2.2	smi_ospf6_bfd.h File Reference	190
2.2.1	Detailed Description	191
2.2.2	Function Documentation	191
2.2.2.1	smi_ospf6_bfd_all_interfaces_set_validate	191
2.2.2.2	smi_ospf6_bfd_all_interfaces_unset_validate	192
2.2.2.3	smi_ospf6_if_bfd_disable_set	192
2.2.2.4	smi_ospf6_if_bfd_disable_unset	192
2.2.2.5	smi_ospf6_if_bfd_set	193
2.2.2.6	smi_ospf6_if_bfd_unset	193
2.2.2.7	smi_ospf6_vlink_bfd_all_set_sdkapi	194
2.2.2.8	smi_ospf6_vlink_bfd_all_unset_sdkapi	194
2.2.2.9	smi_ospf6_vlink_bfd_set	194
2.2.2.10	smi_ospf6_vlink_bfd_unset	195

Chapter 1

File Index

1.1 File List

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

smi_ospf6.h (Provides API for managing OSPF)	3
smi_ospf6_bfd.h (Provides APIs for managing Bidirectional Forwarding De- tection(BFD) in ZebOS)	190

Chapter 2

File Documentation

2.1 smi_ospf6.h File Reference

Provides API for managing OSPF. `#include "smi_client.h"`

`#include "smi_ospf6_msg.h"`

`#include "smi_ospf6_bfd.h"`

Defines

- `#define SMI_OSPF6_VRF_LENGTH 255`
- `#define SMI_OSPF6_TAG_LENGTH 255`
- `#define SMI_OSPF6_RT_MAP_NAME_SIZE 255`
- `#define SMI_CSPF6_DEFAULT_RETRY_INTERVAL_MIN 0`
- `#define SMI_CSPF6_DEFAULT_RETRY_INTERVAL_MAX 3600`
- `#define SMI_OSPF_AREA_ID_FORMAT_ADDRESS 1`
- `#define SMI_OSPF_AREA_ID_FORMAT_DECIMAL 2`
- `#define SMI_OSPF6_CONFIG_LOG_ADJACENCY_CHANGES (1 << 8)`
- `#define SMI_OSPF6_CONFIG_LOG_ADJACENCY_DETAIL (1 << 9)`
- `#define SMI_CSPF_TIE_BREAK_RANDOM 0`
- `#define SMI_CSPF_TIE_BREAK_MOST_FILL 1`
- `#define SMI_CSPF_TIE_BREAK_LEAST_FILL 2`
- `#define SMI_OSPF6_IF_INSTANCE_ID_DEFAULT 0`
- `#define SMI_OSPF6_IF_INSTANCE_ID_MIN 0`
- `#define SMI_OSPF6_IF_INSTANCE_ID_MAX 255`

Functions

- `int smi_ospf6_log_adjacency_changes_set (struct smiclient_globals *azg, u_int32_t vrfId, char *ospf6ProcessTag, u_int16_t ospf6logAdjacencyConfig)`

This function configures the log the adjacency changes.

- int [smi_ospf6_log_adjacency_changes_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int16_t ospf6logAdjacencyConfig)

This function unconfigures the configured logging of adjacency changes.

- int [smi_ospf6_if_mtu_ignore_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)

This function unconfigures the ospfv3 process mtu ignorance of during database description exchange.

- int [smi_ospf6_if_mtu_ignore_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)

This function sets ospfv3 process not to check mtu size during database description exchange.

- int [smi_ospf6_process_shutdown_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function unconfigures the ospfv3 process shutdown configuration.

- int [smi_ospf6_process_shutdown_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function configures the ospfv3 process to shutdown.

- int [smi_ospf6_area_stub_no_summary_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat)

This function makes an area a normal area.

- int [smi_ospf6_if_ipv6_router_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, struct pal_in4_addr areaId, int ospf6AreaFormat, char *ospf6ProcessTag, int instanceId)

This function enables OSPF routing on the specified interface.

- int [smi_ospf6_if_ipv6_router_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, struct pal_in4_addr areaId, char *ospf6ProcessTag, int instanceId)

This function disables OSPF routing on a specified interface.

- int [smi_ospf6_if_network_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int ospf6IfType, int instanceId)

This function sets the network type on the specified interface.

- int [smi_ospf6_if_network_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)

This function removes the network type on the specified interface.

- int [smi_ospf6_if_dead_interval_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t deadInterval, int instanceId)
This function set the interval during which no hello packets are received and after which a neighbor is declared dead. Dead-interval is advertised in the Hello packets. When receiving Hello packets, OSPF router compares dead-interval in a receiving packet and the dead-interval configured on the receiving interface.
- int [smi_ospf6_if_dead_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
This function resets the Router Dead Interval of the specified interface to the default value: 40 seconds.
- int [smi_ospf6_if_hello_interval_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t helloInterval, int instanceId)
This function configures Hello Interval on the specified interface.
- int [smi_ospf6_if_hello_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
This function resets the configured Hello interval on a specified interface to the default value: 10 seconds.
- int [smi_ospf6_if_transmit_delay_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t transmitDelay, int instanceId)
This function sets the estimated time it takes to transmit a Link State Update packet over the interface.
- int [smi_ospf6_if_transmit_delay_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
This function resets the configured OSPF transmit delay of the specified interface to the default value: 1.
- int [smi_ospf6_if_retransmit_interval_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t retransmitInterval, int instanceId)
This function sets the interval between retransmission of Link State Update packets for adjacencies belonging to the interface.
- int [smi_ospf6_if_retransmit_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
This function resets the interval between retransmission of Link State Update packets for adjacencies belonging to the interface to the default value: 5 seconds.
- int [smi_ospf6_if_cost_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t ospf6IfCost, int instanceId)
This function sets the current interface output ospf6IfCost. The configuration is stored regardless of whether or not a real interface exists.
- int [smi_ospf6_if_cost_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)

This function resets the configured output ospf6IfCost on the specified interface to the default value: 10.

- int **smi_ospf6_if_priority_set** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int ospf6IfPriority, int instanceId)

This function configures router priority on the specified interface.

- int **smi_ospf6_if_priority_unset** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)

This function sets the router priority of the specified interface to default value 0.

- int **smi_ospf6_if_disable_all_set** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

This function configures the OSPFv3 routers interconnecting to non-broadcast networks.

- int **smi_ospf6_if_disable_all_unset** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)

This function removes the OSPFv3 routers configuration: interconnecting to non-broadcast networks.

- int **smi_ospf6_if_link_lsa_suppression_set** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int ospf6LsaSuppressionValue, int instanceId)

This function configures the OSPFv3 routers interconnecting to non-broadcast networks.

- int **smi_ospf6_if_te_metric_set** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t ospf6TeMetric, int instanceId)

This function sets the specified metric for OSPFv3 traffic engineering on an interface.

- int **smi_ospf6_if_te_metric_unset** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)

This function unsets a TE metric for OSPFv3 traffic engineering on an interface.

- int **smi_ospf6_restart_graceful_set** (struct smiclient_globals *azg, u_int32_t vrId, u_int16_t gracePeriod, int ospf6RestartGracefulReason)

This function causes a graceful restart of the router. After this command executes, the router immediately shuts down and notifies NSM that the shut down was graceful.

- int **smi_ospf6_process_set_vrf_skddapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *vrfName)
- int **smi_ospf6_process_set_vrf_skddapi** (struct smiclient_globals *azg, u_int32_t vrId, char *vrfName)
- int **smi_ospf6_process_unset_vrf_skddapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *vrfName)
- int **smi_ospf6_process_unset_vrf_skddapi** (struct smiclient_globals *azg, u_int32_t vrId, char *vrfName)

- int **smi_ospf6_hitless_restart_helper_policy_unset_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int policy)
- int **smi_ospf6_hitless_restart_helper_policy_unset** (struct smiclient_globals *azg, u_int32_t vr_id, int policy)
- int **smi_ospf6_restart_graceful_reason_set_wrapper** (struct smiclient_globals *azg, u_int32_t vr_id, u_int32_t ospf6RestartGracefulReason)
- int **smi_ospf6_graceful_restart_hitless_set** (struct smiclient_globals *azg, u_int32_t vrId, u_int16_t gracePeriod, u_int32_t hitlessRestartGraceFul)
- int **smi_ospf6_graceful_restart_hitless_reason_set** (struct smiclient_globals *azg, u_int32_t vrId, u_int32_t ospf6RestarthitlessReason)
- int **smi_ospf6_graceful_restart_hitless_unset** (struct smiclient_globals *azg, u_int32_t vrId, u_int16_t gracePeriod, u_int32_t hitlessRestart)
- int **smi_ospf6_graceful_restart_hitless_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, u_int16_t gracePeriod, u_int32_t hitlessRestart)

- int **smi_ospf6_graceful_restart_set** (struct smiclient_globals *azg, u_int32_t vrId, int gracePeriod, int ospf6RestartGracefulReason)
This function sets the grace period(in seconds) for restarting the router.

- int **smi_ospf6_graceful_restart_unset** (struct smiclient_globals *azg, u_int32_t vrId)
This function sets the grace period(in seconds) for restarting the router.

- int **smi_ospf6_restart_helper_policy_set** (struct smiclient_globals *azg, u_int32_t vrId, int ospf6HelperPolicy)
This function configures the specified helper policy.

- int **smi_ospf6_hitless_restart_helper_never_policy_set** (struct smiclient_globals *azg, u_int32_t vrId, u_int32_t hitlessRestartHelpNever)
- int **smi_ospf6_hitless_restart_helper_never_policy_unset** (struct smiclient_globals *azg, u_int32_t vrId, u_int32_t hitlessRestartHelpNever)
- int **smi_ospf6_hitless_restart_helper_policy_set** (struct smiclient_globals *azg, u_int32_t vr_id, int ospf6HiltlessHelperPolicy)
- int **smi_ospf6_restart_helper_policy_unset** (struct smiclient_globals *azg, u_int32_t vrId, int ospf6HelperPolicy)
This function resets the configuration to default.

- int **smi_ospf6_restart_helper_grace_period_set** (struct smiclient_globals *azg, u_int32_t vrId, int gracePeriod)
This function sets the configured maximum grace period for a neighbor to act as helper.

- int **smi_ospf6_restart_helper_never_router_id_set** (struct smiclient_globals *azg, u_int32_t vrId, struct pal_in4_addr routerId)
This function adds the specified router ID to the Never-Router ID list.

- int **smi_ospf6_restart_helper_never_router_id_unset** (struct smiclient_globals *azg, u_int32_t vrId, struct pal_in4_addr routerId_ptr)

This function deletes the specified router ID to the Never-Router ID list.

- int [smi_ospf6_restart_helper_never_router_unset_all](#) (struct smiclient_globals *azg, u_int32_t vrId)

This function deletes entire Never-Router ID list.

- int [smi_ospf6_restart_helper_grace_period_unset](#) (struct smiclient_globals *azg, u_int32_t vrId)

This function reverts the configured maximum grace period for a neighbor to act as helper to the default value as 1.

- int [smi_ospf6_restart_helper_policy_unset_all](#) (struct smiclient_globals *azg, u_int32_t vrId)

This function resets all configured policies.

- int [smi_ospf6_capability_restart_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6CapabilityRestartMethod)

This function enables OSPFv3 graceful restart capability.

- int [smi_ospf6_capability_restart_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function disables OSPFv3 graceful restart capability.

- int [smi_ospf6_ipv6_ospf_display_route_single_line_set](#) (struct smiclient_globals *azg, u_int32_t vrId)

This function displays IPv6 routing table for OSPFv3 in a single line.

- int [smi_ospf6_ipv6_ospf_display_route_single_line_unset](#) (struct smiclient_globals *azg, u_int32_t vrId)

This function displays IPv6 routing table for OSPFv3 in multiple lines.

- int [smi_ospf6_router_id_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr routerId)

This function configures the OSPFv3 router ID.

- int [smi_ospf6_router_id_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function forces the OSPFv3 to stop routing functionality.

- int [smi_ospf6_address_family_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_char addFamily)

This function sets family mode to IPv4 unicast or IPv6 unicast.

- int [smi_ospf6_address_family_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_char addFamily)

This function sets family mode to IPv4 unicast or IPv6 unicast.

- int [smi_ospf6_abr_type_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int abrType)
This function sets area border router(ABR) type.
- int [smi_ospf6_abr_type_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function sets area border router(ABR) type to OSPF_ABR_TYPE_CISCO.
- int [smi_ospf6_timers_spf_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t minDelay, u_int32_t maxDelay)
This function sets the minimum and maximum delay between a topology change, being either received in an LSA(Link-state advertisement) or self detected and the SP(Shortest Path First)F being run.
- int [smi_ospf6_timers_spf_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function sets SPF minimum and maximum delay to their default values. Minimum default delay is 500 milliseconds and Maximum default delay is 50000 milliseconds.
- int [smi_ospf6_auto_cost_reference_bandwidth_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t ospf6ReferenceBandWidth)
This function sets the reference bandwidth value. OSPF calculates the OSPF metric for an interface by dividing the reference bandwidth.
- int [smi_ospf6_auto_cost_reference_bandwidth_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function sets the reference bandwidth to its default value (100000 kbps).
- int [smi_ospf6_max_concurrent_dd_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int16_t maxDBDescriptor)
This function sets the maximum number of concurrently processed database descriptors.
- int [smi_ospf6_max_concurrent_dd_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function sets the maximum concurrent database descriptors to its default value as 5.
- int [smi_ospf6_max_unuse_packet_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t maxUnusePkt)
This function sets the maximum number of unused OSPFv3 packets.
- int [smi_ospf6_max_unuse_packet_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function sets the maximum number of unused OSPFv3 packets to its default as (200-OSPF6_PACKET_UNUSE_MAX_DEFAULT).

- int [smi_ospf6_max_unuse_lsa_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t maxUnuseLsa)
This function sets the maximum number of unused Link-state advertisement (LSA) packets.
- int [smi_ospf6_max_unuse_lsa_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function unsets the maximum number of unused Link-state advertisement (LSA) packets to default as (200-OSPF6_LSA_UNUSE_MAX_DEFAULT).
- int [smi_ospf6_area_format_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat)
This function sets the format of area ID.
- int [smi_ospf6_area_stub_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat)
This function makes an area a normal area.
- int [smi_ospf6_area_stub_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
This function unsets area as stub configuration.
- int [smi_ospf6_area_no_summary_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
This function configures an area as stub no-summary: inter-area routes are not injected into stub.
- int [smi_ospf6_area_default_cost_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6DefaultCost, int ospf6AreaFormat)
This function configures the ospf6DefaultCost for default summary route sent into a stub area.
- int [smi_ospf6_area_default_cost_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
This function removes the assigned stub default ospf6DefaultCost.
- int [smi_ospf6_area_range_ipv4_set_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat, struct pal_in4_addr ipv4Addr, int prefixlen, int ospf6AreaRangeStatus)
This function configures an OSPF address range.
- int [smi_ospf6_area_range_ipv4_unset_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr ipv4Addr, int prefixlen)
This function removes the configured area range.

- int [smi_ospf6_area_nssa_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat)
This function sets an area as an NSSA (Not-So-Stubby-Area).
- int [smi_ospf6_area_nssa_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
This function unsets an area as an NSSA (Not-So-Stubby-Area).
- int [smi_ospf6_area_nssa_translator_role_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, u_char ospf6AreaNssaTransRole)
This function sets translator role for an NSSA (Not-So-Stubby-Area).
- int [smi_ospf6_area_nssa_translator_role_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
This function removes translator role for an NSSA (Not-So-Stubby-Area).
- int [smi_ospf6_area_nssa_default_originate_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
This function originates Type-7 default LSAs into an NSSA area.
- int [smi_ospf6_area_nssa_default_originate_metric_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6DefaultOriginMetric)
This function originates Type-7 default LSAs with a specific metric into an NSSA area.
- int [smi_ospf6_area_nssa_default_originate_metric_type_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6DefaultOriginMetricType)
This function originates Type-7 default LSAs with a specific metric type into an NSSA area.
- int [smi_ospf6_area_nssa_default_originate_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
This function stops originating Type-7 default LSAs into an NSSA area.
- int [smi_ospf6_area_nssa_no_redistribution_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
This function redistributes routes into an NSSA area.
- int [smi_ospf6_area_nssa_no_redistribution_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
This function stops redistributing routes into an NSSA area.
- int [smi_ospf6_area_nssa_stability_interval_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, u_int32_t stabilityInterval)

This function sets the stability interval for an NSSA area.

- int [smi_ospf6_vlink_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int ospf6AreaFormat)

This function configures the virtual link as specified.

- int [smi_ospf6_vlink_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)

This function removes configuration of the virtual link as specified.

- int [smi_ospf6_vlink_dead_interval_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkDeadInterval)

This function configures the Router Dead Interval of the specified virtual link.

- int [smi_ospf6_vlink_dead_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)

This function unsets the configured Router Dead Interval of the specified virtual link.

- int [smi_ospf6_vlink_hello_interval_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkHelloInterval)

This function configures Hello Interval on the specified virtual link.

- int [smi_ospf6_vlink_hello_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)

This function resets the Hello Interval on a specified virtual link to the default value as 10 seconds.

- int [smi_ospf6_vlink_retransmit_interval_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkRetransmitInterval)

This function sets the retransmit interval on the specified virtual link.

- int [smi_ospf6_vlink_retransmit_interval_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)

This This function resets the configured retransmit interval of the specified virtual link to the default value as 5.

- int [smi_ospf6_vlink_transmit_delay_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkTransmitDelay)

This function configures the transmit delay on a specified virtual link.

- int [smi_ospf6_vlink_transmit_delay_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)

This function unsets the configured transmit delay on a specified virtual link.

- int [smi_ospf6_vlink_instance_id_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int VirtualLinkInstanceId)

This function configures the instance ID on the specified virtual link.

- int [smi_ospf6_vlink_instance_id_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)

This function resets the configured instance ID of the specified virtual link to the default value as 0.

- int [smi_ospf6_capability_cspf_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function enables the CSPF capability for an OSPFv3 process.

- int [smi_ospf6_capability_cspf_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function disables the CSPF capability for an OSPFv3 process.

- int [smi_ospf6_enable_db_summary_opt](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function enables the OSPFv3 Database Summary List optimization.

- int [smi_ospf6_disable_db_summary_opt](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function disables the OSPFv3 Database Summary List optimization.

- int [smi_ospf6_redistribute_set_by_ospf_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)

This function redistributes routes from the specified protocol.

- int [smi_ospf6_redistribute_set_by_ospf6_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspf6Tag)

This function redistributes routes from the specified protocol.

- int [smi_ospf6_redistribute_unset_by_ospf6_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspf6Tag)

This function stops redistributing routes from the specified protocol.

- int [smi_ospf6_redistribute_unset_by_ospf_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)

This function stops redistributing routes from the specified protocol.

- int [smi_ospf6_redistribute_metric_type_set_by_ospf_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag, u_char ospf6RedistMetricType)

This function sets the metric type for a redistributed route.

- int [smi_ospf6_redistribute_metric_type_set_by_ospf6_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag, u_char ospf6RedistMetricType)

This function sets the metric type for a redistributed route.

- int [smi_ospf6_redistribute_metric_type_unset_by_ospf_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)

This function sets the metric type of a redistributed route to its default value 2.

- int [smi_ospf6_redistribute_metric_type_unset_by_ospf6_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag)

This function sets the metric type of a redistributed route to its default value 2.

- int [smi_ospf6_redistribute_metric_set_by_ospf_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag, u_int32_t ospf6RedistMetricValue)

This function sets the metric for a redistributed route.

- int [smi_ospf6_redistribute_metric_set_by_ospf6_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag, u_int32_t ospf6RedistMetricValue)

This function sets the metric for a redistributed route.

- int [smi_ospf6_redistribute_metric_unset_by_ospf_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)

This function sets the metric of a redistributed route to its default value 16777215.

- int [smi_ospf6_redistribute_metric_unset_by_ospf6_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag)

This function sets the metric of a redistributed route to its default value 16777215.

- int [smi_ospf6_redistribute_route_tag_set_by_ospf_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag, u_int32_t ospf6RedistTagVal)

This function sets the metric for a redistributed route.

- int [smi_ospf6_redistribute_route_tag_set_by_ospf6_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspf6Tag, u_int32_t ospf6RedistTagVal)

This function sets the metric for a redistributed route.

- int [smi_ospf6_redistribute_route_tag_unset_by_ospf_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)

This function sets the metric of a redistributed route to its default value 16777215.

- int [smi_ospf6_redistribute_route_tag_unset_by_ospf6_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspf6Tag)

This function sets the metric of a redistributed route to its default value 16777215.

- int [smi_ospf6_default_metric_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t ospf6DefaultMetric)

This function sets default metric values for the OSPFv3 routing protocol.

- int [smi_ospf6_default_metric_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function unsets configured default metric.

- int [smi_ospf6_routemap_set_by_ospf_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag, char *routeMapName)

This function assigns a route-map to a redistributed protocol.

- int [smi_ospf6_routemap_set_by_ospf6_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspf6Tag, char *routeMapName)

This function assigns a route-map to a redistributed protocol.

- int [smi_ospf6_routemap_unset_by_ospf_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)

This function removes a route map from a redistributed protocol.

- int [smi_ospf6_routemap_unset_by_ospf6_src_tag_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspf6Tag)

This function removes a route map from a redistributed protocol.

- int [smi_ospf6_routemap_default_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *routeMapName)
This function sets default route map from a redistributed protocol.
- int [smi_ospf6_routemap_default_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function unsets default route map from a redistributed protocol.
- int [smi_ospf6_redist_default_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistDefaultOrigin)
This function sets default redistributed protocol table as from default information originate.
- int [smi_ospf6_redist_default_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function unsets the default redistributed protocol table as from default information originate.
- int [smi_ospf6_router_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function initiates the OSPFv3 router instance.
- int [smi_ospf6_router_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function removes the OSPFv3 router instance.
- int [smi_ospf6_passive_if_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, char *ospf6ProcessTag)
This function configures the specified interface into passive mode.
- int [smi_ospf6_passive_if_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, char *ospf6ProcessTag)
This function unsets the passive interface configuration.
- int [smi_ospf6_passive_if_default_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function sets all the interfaces into passive mode.
- int [smi_ospf6_passive_if_default_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
This function removes all the interfaces from passive and no passive interface list.
- int [smi_ospf6_distance_all_set](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6AdminDistance)
This function sets OSPFv3 route administrative distances based on route type. It sets distance for an entire group of routes rather than a specific route that passes an access list.

- int **smi_ospf6_distance_set_wrap_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int administrativeDistance)
- int **smi_ospf6_distance_set_wrap** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int administrativeDistance)
- int **smi_ospf6_distance_all_unset** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function unsets OSPFv3 route administrative distances based on route type.

- int **smi_ospf6_distance_intra_area_set** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int intraDistance)

This function sets distance for intra-area routes.

- int **smi_ospf6_distance_intra_area_unset** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function unsets distance for intra-area routes.

- int **smi_ospf6_distance_inter_area_set** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int interDistance)

This function sets distance for inter-area routes.

- int **smi_ospf6_distance_inter_area_unset** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function unsets distance for inter-area routes.

- int **smi_ospf6_distance_external_set** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int externalDistance)

This function sets distance for external routes.

- int **smi_ospf6_distance_external_unset** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

This function unsets distance for external routes.

- int **smi_ospf6_distance_unset_wrap** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, int administrativeDistance)

- int **smi_ospf6_distribute_list_in_set** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *accessListName)

This function sets access-list to filter networks for incoming routing updates. It redistributes other routing protocols into the OSPF routing table.

- int **smi_ospf6_distribute_list_in_unset** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *accessListName)

This function unsets access-list to filter networks for incoming routing updates.

- int **smi_ospf6_distribute_list_out_set** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *ospf6DistributeTag, char *accessListName)

This function sets access-list and redistributed protocol type filter networks for outgoing routing updates. It redistributes other routing protocols into the OSPF routing table.

- int [smi_ospf6_distribute_list_out_unset](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *stag, char *accessListName)

This function unsets access-list and redistributed protocol type filter networks for outgoing routing updates.

- int [smi_ospf6_summary_address_ipv4_set_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)

This function summarizes external routes with the specified address range.

- int [smi_ospf6_summary_address_ipv4_unset_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)

This function removes a summary address.

- int [smi_ospf6_summary_address_tag_ipv4_set_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen, u_int32_t routeTag)

This function sets a ospf6ProcessTag value to use as a "match" value for controlling redistribution via route maps.

- int [smi_ospf6_summary_address_tag_ipv4_unset_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)

This function removes a tag value to use as a "match" value for controlling redistribution via route maps.

- int [smi_ospf6_summary_address_not_advertise_ipv4_set_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)

This function suppresses external routes that match a specified address range.

- int [smi_ospf6_summary_address_not_advertise_ipv4_unset_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)

This function stops suppressing external routes that match a specified address range.

- int [smi_ospf6_summary_address_not_advertise_ipv6_unset_sdkapi](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in6_addr ipv6Addr, u_char masklen)

This function stops suppressing external routes that match a specified address range.

- int [smi_ospf6_get_router_id](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr *routerId)

This function gets 32-bit IPv4 interface address uniquely identifying the router in the Autonomous System.

- int [smi_ospf6_get_admin_stat](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *adminStat)

This function gets administrative status of OSPFv3 in the router. The value 'enabled' denotes that the OSPFv3 Process is active on at least one.

- int [smi_ospf6_get_version_num](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *versionNum)

This function gets the version number of OSPF for IPv6.

- int [smi_ospf6_get_area_bdr_rtr_status](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *areaBdrRtrStatus)

This function gets the flag to denote whether this router is an area border router.

- int [smi_ospf6_get_asbdr_rtr_status](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *asBdrRtrStatus)

This function gets flag to note whether this router is configured as an Autonomous System border router.

- int [smi_ospf6_get_as_scope_lsa_count](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *asScopeLsaCount)

This function gets the number of (Autonomous System)AS-Scope (e.g. AS-External) link state advertisements in the link state database.

- int [smi_ospf6_get_as_scope_lsa_cksumsum](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *asScopeLsaCksumsum)

This function gets the 32-bit unsigned sum of the LS(Link-state) checksums of the AS-scoped link state advertisements contained in the link state database.

- int [smi_ospf6_get_originate_new_lsas](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *originateNewLsas)

This function gets the number of new link-state advertisements that have been originated. This number is incremented each time the router originates a new LSA.

- int [smi_ospf6_get_rx_new_lsas](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *rxNewLsas)

This function gets the number of link state advertisements received determined to be new instantiations. This number does not include newer instantiations of self-originated link state advertisements.

- int [smi_ospf6_get_extern_lsa_count](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *externLsaCount)

This function gets the number of External(LS(Link-state) type 0x4005) in the link state database.

- int [smi_ospf6_get_reference_bandwidth](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *referenceBandwidth)

This function gets reference bandwidth in kilobits per second for calculating default interface metrics.

- int [smi_ospf6_get_restart_support](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *restartSupport)

This function gets the value that indicates whether the router supports OSPF graceful restart.

- int [smi_ospf6_get_restart_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *restartInterval)

This function gets the interval of the graceful restart timeout for the router.

- int [smi_ospf6_get_restart_strict_lsa_check](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *restartStrictLsaCheckStatus)

This function gets the value that indicates whether strict LSA checking is enabled for graceful restart on the router.

- int [smi_ospf6_get_restart_status](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *restartStatus)

This function gets the current status of OSPF graceful restart capability for the router.

- int [smi_ospf6_get_restart_age](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *restartAge)

This function gets remaining time, in seconds, in the current OSPF graceful restart interval for the router.

- int [smi_ospf6_get_restart_exit_reason](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *restartExitReason)

This function gets the outcome of the last attempt at a graceful restart for the router.

- int [smi_ospf6_get_notification_enable](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *notificationEnable)

This function gets the value that indicates whether the generation of OSPFv3 notifications is enabled.

- int [smi_ospf6_get_stub_router_support](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *stubRouterSupport)

This function gets the value that indicates whether the router supports the stub router functionality.

- int [smi_ospf6_get_stub_router_advertisement](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *stubRouterAdvertisement)

This function gets the value that indicates whether the router advertises stub or standard LSAs(Link state advertisements).

- int [smi_ospf6_get_discontinuity_time](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *discontinuityTime)

This function gets the value of sysUpTime counter, which is the most recent time at which any one of this MIBs counters experienced a discontinuity.

- int [smi_ospf6_get_restart_time](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int *restartTime)

This function gets the value of sysUpTime on the most recent time at which the ospfv3RestartExitReason object was updated.

- int [smi_ospf6_set_asbdr_rtr_status](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ospf6AsbdrRtrStatus)

This function sets the router as an Autonomous System (AS) border router.

- int [smi_ospf6_get_import_as_extern](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *importAsExtern)

This function indicates whether an area is a stub area, NSSA, or standard area.

- int [smi_ospf6_get_spf_runs](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *spf_runs)

This function gets number of times that the intra-area route table has been calculated using this area's link state database.

- int [smi_ospf6_get_area_bdr_rtr_count](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaBdrRtrCount)

This function gets total number of area border routers reachable within this area. This is initially zero, and is calculated in each Shortest Path First (SPF) pass.

- int [smi_ospf6_get_asbdr_rtr_count](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *asbdrRtrCount)

This function gets the total number of Autonomous System border routers reachable within this area. This is initially zero, and is calculated in each SPF pass.

- int [smi_ospf6_get_area_lsa_count](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaLsaCount)

This function gets the total number of Area-scope link state advertisements in this area's link state database.

- int [smi_ospf6_get_area_lsa_cksum_sum](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaLsaChecksumSum)

This function gets the 32-bit unsigned sum of the Area-scope link state advertisements' LS checksums contained in this area's link state database. The sum can be used to determine if there has been a change in a router's link state database or to compare the link state database of two routers.

- int [smi_ospf6_get_area_summary](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaSummary)

This function gets the variable ospfv3AreaSummary controls the import of Inter-Area LSAs into stub and NSSA areas.

- int [smi_ospf6_get_area_status](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaStatus)

This function gets object which permits management of the table by facilitating actions such as row creation, construction, and destruction.

- int [smi_ospf6_get_stub_metric](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *stubMetric)

This function gets the metric value advertised for the default route into stub and NSSA areas. By default, this equals the least metric among the interfaces to other areas.

- int [smi_ospf6_get_area_nssa_trans_role](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaNssaTransRole)

This function indicates an NSSA border router's policy to perform NSSA(Not-so-Stubby Area) translation of NSSA-LSAs into(Autonomous System) AS-External-LSAs.

- int [smi_ospf6_get_area_nssa_trans_state](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaNssaTransState)

This function indicates if and how an NSSA border router is performing NSSA(Not-so-Stubby Area) translation of NSSA-LSAs into AS-External-LSAs.

- int [smi_ospf6_get_area_nssa_trans_stability_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaNssaTransStabilityInterval)

This function gets the stability interval defined as the number of seconds after an elected translator determines its services are no longer required that it should continue to perform its translation duties.

- int [smi_ospf6_get_area_nssa_trans_events](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaNssaTransEvents)

This function indicates the number of Translator state changes that have occurred since the last start-up of the OSPFv3 routing process.

- int [smi_ospf6_get_area_stub_metric_type](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaStubMetricType)

This function gets the type of metric advertised as a default route.

- int [smi_ospf6_get_area_te_enabled](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaTeEnabled)

This function indicates whether or not traffic engineering is enabled in the area.

- int [smi_ospf6_set_area_nssa_trans_role](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, u_int32_t ospf6AreaNssaTransRole)

This function sets an NSSA border router's policy to perform NSSA(Not-so-Stubby Area) translation of NSSA-LSAs into(Autonomous System) AS-External-LSAs.

- int [smi_ospf6_set_area_stub_metric_type](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6StubMetricType)

This function sets the type of metric advertised as a default route.

- int [smi_ospf6_set_area_te_enabled](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6TeStatus)

This function sets to whether or not traffic engineering is enabled in the area.

- int [smi_ospf6_get_as_lsdb_sequence](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *asLsdbSequence)

This function gets the sequence number field is a signed 32-bit integer. It is used to detect old and duplicate link state advertisements.

- int [smi_ospf6_get_as_lsdb_age](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *asLsdbAge)

This function gets the age of the link state advertisement in seconds.

- int [smi_ospf6_get_as_lsdb_checksum](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *asAsdbChecksum)

This function gets the checksum of the complete contents of the advertisement, excepting the age field. The age field is excepted so that an advertisement's age can be incremented without updating the checksum.

- int [smi_ospf6_get_as_lsdb_advertisement](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, struct smi_ospf6_lsa_header *asLsdbAdvertisement, size_t *size)

This function gets the entire link state advertisement including the header.

- int [smi_ospf6_get_as_lsdb_type_known](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *asLsdbTypeKnown)

This function gets value which indicates that the LSA(Link state advertisement) type is recognized by this router.

- int [smi_ospf6_get_area_lsdb_sequence](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *areaLsdbSequence)

This function gets the sequence number field in Link state advertisement, is a signed 32-bit integer. It is used to detect old and duplicate link state advertisements.

- int [smi_ospf6_get_area_lsdb_age](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *areaLsdbAge)

This function gets age field in link state advertisement which is the age of the link state advertisement in seconds.

- int [smi_ospf6_get_area_lsdb_checksum](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *areaLsdbChecksum)

This function gets checksum field which is the computed checksum of the complete contents of the link state advertisement, excepting the age field. The age field is excepted so that an advertisement's age can be incremented without updating the checksum.

- int [smi_ospf6_get_area_lsdb_advertisement](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, struct smi_ospf6_lsa_header *lsa, size_t *size)

This function gets the entire link state advertisement, including its header.

- int [smi_ospf6_get_area_lsdb_type_known](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *areaLsdbTypeKnown)

This function gets the value which indicates that the LSA(Link state advertisement) type is recognized by this router.

- int [smi_ospf6_get_link_lsdb_sequence](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *linkLsdbSequence)

This function gets the LSDB sequence number of the local Link-Scope LSA for the given interface. The sequence number is used to detect old and duplicate LSAs.

- int [smi_ospf6_get_link_lsdb_age](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *linkLsdbAge)

This function gets the age of the LSA that is stored in the local Link-Scope LSDB.

- int [smi_ospf6_get_link_lsdb_checksum](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *linkLsdbChecksum)

This function gets the checksum of the contents of the LSA that is stored in the specified local Link-Scope LSDB, excluding the age field, for the given OSPFv3 interface.

- int [smi_ospf6_get_link_lsdb_advertisement](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, struct smi_ospf6_lsa_header *lsa, size_t *size)

This function gets the entire LSA, including its header, from the local Link-Scope LSDB for the given interface.

- int [smi_ospf6_get_link_lsdb_type_known](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *linkLsdbTypeKnown)

This function gets the value that indicates whether the LSA type is recognized by the given router.

- int [smi_ospf6_get_if_area_id](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr *ifAreaId)
This function gets the area ID, which is a 32-bit integer uniquely identifying the area to which the given interface connects. Area ID 0 is used for the OSPFv3 backbone.
- int [smi_ospf6_get_if_type](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifType)
This function gets the interface type of the given OSPFv3 interface.
- int [smi_ospf6_get_if_admin_stat](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifAdminStat)
This function gets the administrative status of the given OSPFv3 interface.
- int [smi_ospf6_get_if_rtr_priority](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifRtrPriority)
This function gets the priority value of the given interface. A value of 0 signifies that the router is not eligible to become the Designated Router on this particular network. In the event of a tie in this value, the routers use their Router identifier as a tie breaker.
- int [smi_ospf6_get_if_transit_delay](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifTransitDelay)
This function gets the transit-delay value of the given OSPFv3 interface. This value is an estimate of the number of seconds required to transmit a link-state update packet through the interface.
- int [smi_ospf6_get_if_retrans_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifRetransInterval)
This function gets the number of seconds between LSA retransmissions for adjacencies belonging to the given interface.
- int [smi_ospf6_get_if_hello_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifHelloInterval)
This function gets the length of time, in seconds, between the Hello packets that the router sends on the given interface. This value must be the same for all routers attached to a common network.
- int [smi_ospf6_get_if_rtr_dead_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifRtrDeadInterval)
This function gets the number of seconds that a router's Hello packets have not been seen before its neighbors declare the router down on the given interface.
- int [smi_ospf6_get_if_poll_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifPollInterval)
This function gets the larger time interval, in seconds, between the Hello packets sent to an inactive, non-broadcast multi-access neighbor for the given interface.
- int [smi_ospf6_get_if_state](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifState)

This function gets the state of the given OSPFv3 interface.

- int [smi_ospf6_get_if_dr](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr *ifDr)

This function gets the router identifier of the Designated Router for the given OSPFv3 interface.

- int [smi_ospf6_get_if_bdr](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr *ifBdr)

This function gets the interface address of the Backup Designated Router (BDR).

- int [smi_ospf6_get_if_events](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifEvents)

This function gets the number of times the given OSPFv3 interface has changed its state or an error has occurred.

- int [smi_ospf6_get_if_status](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifStatus)

This function gets the status of whether OSPFv3 is configured on the given interface.

- int [smi_ospf6_get_if_demand](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifDemand)

This function gets the value that indicates whether Demand OSPFv3 procedures (Hello suppression to FULL neighbors and setting the DoNotAge flag on propagated LSAs) are performed on the given interface.

- int [smi_ospf6_get_if_metric_value](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifMetricValue)

This function gets the metric value assigned to the given interface.

- int [smi_ospf6_get_if_link_scope_lsa_count](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifLinkScopeLsaCount)

This function gets the total number of Link-Scope LSAs in the given link's LSDB.

- int [smi_ospf6_get_if_link_lsa_cksumsum](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifLinkLsaCksumsum)

This function gets the 32-bit unsigned sum of the Link-Scope LSAs' LS checksums contained in the given link's LSDB.

- int [smi_ospf6_get_if_demand_nbr_probe](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifDemandNbrProbe)

This function gets the value that indicates whether neighbor probing is enabled to determine whether the neighbor is inactive. If neighbor probing is disabled, the neighbor is inactive.

- int [smi_ospf6_get_if_demand_nbr_probe_retrans_limit](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifDemandNbrProbeRetransLimit)

This function gets the configured number of consecutive LSA retransmissions permitted before the neighbor is considered inactive and the neighbor adjacency is brought down.

- int [smi_ospf6_get_if_demand_nbr_probe_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifDemandNbrProbeInterval)

This function gets the number of times the neighbor is to be probed.

- int [smi_ospf6_get_if_te_disabled](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifTeDisabled)

This function gets the value that indicates whether traffic engineering is disabled on the given interface when traffic engineering is enabled in the area to which the interface is attached.

- int [smi_ospf6_get_if_link_lsa_suppression](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int *ifLinkLsaSuppression)

This function gets the value that indicates whether link LSA origination is suppressed for broadcast or NBMA interface types.

- int [smi_ospf6_set_if_type](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int ospf6IfType)

This function sets the interface type of the given OSPFv3 interface.

- int [smi_ospf6_set_if_admin_stat](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int adminstat)
- int [smi_ospf6_set_if_rtr_dead_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int ifRtrDeadInterval)

This function sets the number of seconds that a router's Hello packets have not been seen on the given interface before its neighbors declare the router down.

- int [smi_ospf6_get_virt_if_index](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtIfIndex)

This call gets the interface ID assigned to this OSPFv3 virtual interface.

- int [smi_ospf6_get_virt_if_instid](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtIfInstanceId)

This function gets the local Interface Instance identifier assigned by the OSPFv3 process to the given OSPFv3 virtual interface.

- int [smi_ospf6_get_virt_if_transmit_delay](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtTfTransmitDelay)

This call gets the estimated number of seconds it takes to transmit a link-state update packet over this virtual interface.

- int [smi_ospf6_get_virt_if_retrans_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtIfRetransInterval)

This call gets the number of seconds between link-state advertisement retransmissions, for adjacencies belonging to this virtual interface.

- int [smi_ospf6_get_virt_if_hello_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtIfHelloInterval)

This call gets the length of time, in seconds, between the Hello packets that the router sends on the virtual interface.

- int [smi_ospf6_get_virt_if_rtr_dead_interval](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtIfRtrDeadInterval)

This call gets the number of seconds that a router's Hello packets have not been seen on this virtual interface, before its neighbors declare the router down.

- int [smi_ospf6_get_virt_if_state](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtIfState)

This call gets the OSPF virtual interface state.

- int [smi_ospf6_get_virt_if_events](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtIfEvents)

This call gets the number of state changes or error events on this Virtual Link.

- int [smi_ospf6_get_virt_if_status](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtIfStatus)

This call gets the status of the specified virtual interface.

- int [smi_ospf6_get_virt_if_link_scope_lsa_count](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtIfLinkScopeLsaCount)

This call gets the total number of Link-Scope link-state advertisements in this virtual link's link-state database.

- int [smi_ospf6_get_virt_if_link_lsa_cksumsum](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtIfLinkLsaChecksumsum)

This call gets the 32-bit unsigned sum of the Link-Scope link-state advertisements' LS checksums contained in this virtual link's link-state database.

- int [smi_ospf6_get_nbr_address_type](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrAddressType)

This call gets the address type of ospfv3NbrAddress.

- int [smi_ospf6_get_nbr_address](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, char *nbrAddress, size_t *size)

This call gets the IPv6 address the neighbor associated with the local link.

- int [smi_ospf6_get_nbr_options](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrOptions)

This call gets a Bit Mask corresponding to the neighbor's options field.

- int [smi_ospf6_get_nbr_priority](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrPriority)

This call gets the priority of this neighbor in the DR election algorithm.

- int [smi_ospf6_get_nbr_state](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrState)

This call gets the state of the relationship with this neighbor.

- int [smi_ospf6_get_nbr_events](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrEvents)

This call gets the number of times this neighbor relationship has changed state, or an error has occurred.

- int [smi_ospf6_get_nbr_lsretransq_len](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrLsretransqLen)

This call gets the current length of the retransmission queue.

- int [smi_ospf6_get_nbr_hello_suppressed](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrHelloSuppressed)

This call gets the indication whether Hellos are being suppressed to the neighbor.

- int [smi_ospf6_get_nbr_if_id](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrIfId)

This call gets the interface ID that the neighbor advertises in its Hello Packets on this link.

- int [smi_ospf6_get_nbr_restart_helper_status](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrRestartHelperStatus)

This function gets the value that indicates whether the given router is a graceful restart helper for the neighbor.

- int [smi_ospf6_get_nbr_restart_helper_age](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrRestartHelperAge)

This function gets the remaining time in the current OSPF graceful restart interval for the given router that is acting as a restart helper for the neighbor.

- int [smi_ospf6_get_nbr_restart_helper_exit_reason](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int *nbrRestartHelperExitReason)

This function gets the outcome of the last attempt the given router was acting as a graceful restart helper for the neighbor.

- int [smi_ospf6_get_virt_nbr_ifindex](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrIfindex)

This function gets the priority number of the neighbor in the Designated Router Election algorithm. This number is used in the Designated Router Election algorithm to determine whether the neighbor is eligible to become the Designated Router on a given network. The value 0 disqualifies the router.

- int [smi_ospf6_get_virt_nbr_ifinstid](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrIfInstanceId)

This function gets the interface instance identifier of the given virtual interface over which the neighbor can be reached.

- int [smi_ospf6_get_virt_nbr_address_type](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrAddressType)

This call gets the address type of ospfv3VirtNbrAddress.

- int [smi_ospf6_get_virt_nbr_address](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, char *virtNbrAddress, size_t *size)

This call gets the IPv6 address of the virtual neighbor associated with the local link. It must be a Site-Local or Global scope address.

- int [smi_ospf6_get_virt_nbr_options](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrOptions)

This call gets a Bit Mask corresponding to the virtual neighbor's options field.

- int [smi_ospf6_get_virt_nbr_state](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrState)

This call gets the state of the relationship with this virtual Neighbor.

- int [smi_ospf6_get_virt_nbr_events](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrEvents)

This call gets the number of times this virtual neighbor has changed its state, or an error has occurred.

- int [smi_ospf6_get_virt_nbr_ls_retransq_len](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrLsRetransqLen)

This call gets the current length of the retransmission queue.

- int [smi_ospf6_get_virt_nbr_hello_suppressed](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrHelloSuppressed)

This call gets the indication whether Hellos are being suppressed to the virtual neighbor.

- int [smi_ospf6_get_virt_nbr_if_id](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrIfId)

This call gets the interface ID that the neighbor advertises in its Hello Packets on this virtual link.

- int [smi_ospf6_get_virt_nbr_restart_helper_status](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrRestartHelperStatus)

This function gets the value that indicates whether the virtual router is acting as a graceful restart helper for the neighbor.

- int [smi_ospf6_get_virt_nbr_restart_helper_age](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrRestartHelperAge)

This function gets the remaining time in the current OSPF graceful restart interval for the given virtual router that is acting as a restart helper for the neighbor.

- int [smi_ospf6_get_virt_nbr_restart_helper_exit_reason](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int *virtNbrRestartHelperExitReason)

This function gets the outcome of the last attempt of the given virtual router at acting as a graceful restart helper for the neighbor.

- int [smi_ospf6_get_virt_link_lsdb_sequence](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int ospf6LsaType, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *virtLinkLsdbSequence)

This function gets the aggregate route tag, which is advertised only in the summarized As-External LSA when summarizing from NSSA-LSAs to AS-External-LSAs.

- int [smi_ospf6_get_virt_link_lsdb_age](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int ospf6LsaType, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *virtLinkLsdbAge)

This function gets the age of the LSA that is stored in the local Link-Scope LSDB for the given virtual OSPFv3 interface.

- int [smi_ospf6_get_virt_link_lsdb_checksum](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int ospf6LsaType, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *virtLinkLsdbChecksum)

This function gets the entire LSA, including its header, from the local Link-Scope LSDB for the OSPFv3 virtual link.

- int [smi_ospf6_get_virt_link_lsdb_advertisement](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int ospf6LsaType, struct pal_in4_addr routerId, struct pal_in4_addr lsId, struct smi_ospf6_lsa_header *virtLinkLsdbAdvertisement, size_t *size)

This function gets the entire LSA, including its header, from the local Link-Scope LSDB for the OSPFv3 virtual link.

- int [smi_ospf6_get_virt_link_lsdb_type_known](#) (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int ospf6LsaType, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int *virtLinkLsdbTypeKnown)

This function gets the value that indicates whether the LSA type is recognized by the given router.

- int [merge_ospf6_list_nbr_info](#) (struct list *listDest, struct sminbrByTagList *listSrc)

This function returns the list of neighbor by tag.

- int [merge_ospf6_list_nbr_detail_info](#) (struct list *listDest, struct sminbrDetailByTagList *listSrc)
- int [merge_ospf6_list_nbr_if_info](#) (struct list *listDest, struct sminbrByIfList *listSrc)
- int [merge_ospf6_list_nbr_detail_if_info](#) (struct list *listDest, struct sminbrDetailByIfList *listSrc)
- int [merge_ospf6_list_nbr_id_info](#) (struct list *listDest, struct sminbrByIdList *listSrc)
- int [smi_ospf6_show_neighbor_by_tag](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *nbrData, int(*funpointer)(struct sminbrByTagList *nbrData))
- int [smi_ospf6_show_neighbor_detail_by_tag](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *nbrData, int(*funpointer)(struct sminbrDetailByTagList *getinfo))

This function returns the list of neighbor detail by tag.

- int [smi_ospf6_show_neighbor_detail_by_interface_name](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ifName, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *nbrData, int(*funpointer)(struct sminbrDetailByIfList *getinfo))

This function returns the list of neighbor info by neighbor interface name.

- int [smi_ospf6_show_neighbor_by_interface_name](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ifName, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *nbrData, int(*funpointer)(struct sminbrByIfList *getinfo))

This function returns the list of neighbor by neighbor interface name.

- int [smi_ospf6_show_neighbor_by_nbr_id](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, char *nbrid, int startIndex, int endIndex, struct list *nbrData, int(*funpointer)(struct sminbrByIdList *getinfo))

This function returns the list of neighbor info by neighbor id.

- int [smi_ospf6_show_neighbor_by_if_nbr_id](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ifName, char *ospf6ProcessTag, char *nbrid, int startIndex, int endIndex, struct list *nbrList, int(*funpointer)(struct list *nbrList))

This function returns the list of neighbor info by interface name and neighbor id.

- int [smi_ospf6_show_neighbor_detail_by_if_nbr_id](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ifName, char *ospf6ProcessTag, char *nbrid, int startIndex, int endIndex, struct list *nbrList, int(*funpointer)(struct list *nbrList))

This function returns the list of neighbor detail info by interface name and neighbor id.

- int [smi_ospf6_show_virtual_link](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *lvirtualLink, int(*funpointer)(struct list *lvirtualLink))

This function returns the list of virtual links.

- int [smi_ospf6_show_virtual_link_brief](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *lvirtualLink, int(*funpointer)(struct list *lvirtualLink))

This function returns the list of virtual links brief info.

- int [smi_ospf6_show_interface_brief](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ifName, int startIndex, int endIndex, struct list *interfaceBrief, int(*funpointer)(struct list *interfaceBrief))

This function returns the list of Interface brief by interface name.

- int [smi_ospf6_show_interface](#) (struct smiclient_globals *azg, u_int32_t vrid, char *ifName, int startIndex, int endIndex, struct list *linterfaceInfo, int(*funpointer)(struct list *linterfaceInfo))

This function returns the list of Interface detail by interface name.

- int [smi_ospf6_show_neighbor_summary_by_tag](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *nbrList, int(*funpointer)(struct list *nbrList))

This function returns the list of neighbor info summary.

- int [smi_ospf6_show_neighbor_summary_by_interface_name](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *nbrList, int(*funpointer)(struct list *nbrList))

This function returns the list of neighbor info summary.

- int [smi_ospf6_show_neighbor_summary_by_nbr_id](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *nbr_id, int startIndex, int endIndex, struct list *nbrList, int(*funpointer)(struct list *nbrList))

This function returns the list of neighbor info summary.

- int [smi_ospf6_show_neighbor_summary_by_if_nbr_id](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, char *ospf6ProcessTag, char *nbrid, int startIndex, int endIndex, struct list *nbrList, int(*funpointer)(struct list *nbrList))

This function returns the list of neighbor info summary.

- int [smi_ospf6_show_interface_by_tag](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *ifName, int startIndex, int endIndex, struct list *interfaceInfo, int(*funpointer)(struct list *interfaceInfo))

This function returns the list of Interface detail by process tag.

- int [smi_ospf6_show_interface_brief_by_tag](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *ifName, int startIndex, int endIndex, struct list *interfaceInfo, int(*funpointer)(struct list *interfaceInfo))

This function returns the list of Interface brief by process tag.

- int [smi_ospf6_show_border_routers](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *brInfo, int(*funpointer)(struct list *brInfo))

This function returns the list of border routers info.

- int [smi_ospf6_show_database](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *dbInfo, int(*funpointer)(struct list *dbInfo))

This function returns the list of process database info.

- int [smi_ospf6_show_route](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *routeInfo, int(*funpointer)(struct list *routeInfo))

This function returns the list of route info.

- int [smi_ospf6_show_route_summary](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *routeInfo, int(*funpointer)(struct list *routeInfo))

This function returns the list of route info summary.

- int [smi_ospf6_show_route_by_prefix](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct prefix_ipv6 prefix, int startIndex, int endIndex, struct list *routeInfo, int(*funpointer)(struct list *routeInfo))

This function returns the list of route info by prefix.

- int [smi_ospf6_show_process](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int startIndex, int endIndex, struct list *procInfo, int(*funpointer)(struct list *procInfo))

This function returns the list of process info.

- int [smi_ospf6_debug](#) (struct smiclient_globals *azg, u_int32_t vrId, int ospf6Debug)

Use this function to specify debugging option for OSPFv3 ZebOS information.

- int [smi_ospf6_no_debug](#) (struct smiclient_globals *azg, u_int32_t vrId, int ospf6Debug)

Use this function to disable debugging option for OSPFv3 ZebOS information.

- int [smi_ospf6_show_debug](#) (struct smiclient_globals *azg, u_int32_t vrId, struct ospf6_debug_show *debug_show)

Use this function to show debugging option for OSPFv3 ZebOS information.

- int [smi_debug_ospf6_packet](#) (struct smiclient_globals *azg, int vrId, int packetType, int ospf6DebugMode, int ospf6DebugDetail)

Use this function to specify the packet debugging options for OSPFv3 ZebOS information.

- int [smi_debug_no_ospf6_packet](#) (struct smiclient_globals *azg, int vrId, int packetType, int ospf6DebugMode, int ospf6DebugDetail)

use this function to disable the packet debugging options for ospfv3 zebos information

- int [smi_ospf6_log_adjacency_changes_set_validate](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int16_t ospf6logAdjacencyConfig)
- int [smi_ospf6_log_adjacency_changes_unset_validate](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int16_t ospf6logAdjacencyConfig)
- int [smi_ospf6_log_adjacency_changes_set_wrapper](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *ospf6logAdjacency)
- int [smi_ospf6_log_adjacency_changes_set_wrapper_validate](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *ospf6logAdjacency)

- int **smi_ospf6_log_adjacency_changes_unset_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *ospf6logAdjacency)
- int **smi_ospf6_log_adjacency_changes_unset_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *ospf6logAdjacency)
- int **smi_ospf6_if_mtu_ignore_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
- int **smi_ospf6_if_mtu_ignore_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
- int **smi_ospf6_process_shutdown_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_process_shutdown_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_area_stub_no_summary_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- int **smi_ospf6_area_stub_no_summary_unset** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- int **smi_ospf6_area_stub_no_summary_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat)
- int **smi_ospf6_if_ipv6_router_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, struct pal_in4_addr areaId, int ospf6AreaFormat, char *ospf6ProcessTag, int instanceId)
- int **smi_ospf6_if_ipv6_router_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, struct pal_in4_addr areaId, char *ospf6ProcessTag, int instanceId)
- int **smi_ospf6_if_network_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int ospf6IfType, int instanceId)
- int **smi_ospf6_if_network_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
- int **smi_ospf6_if_dead_interval_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t deadInterval, int instanceId)
- int **smi_ospf6_if_dead_interval_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
- int **smi_ospf6_if_hello_interval_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t helloInterval, int instanceId)
- int **smi_ospf6_if_hello_interval_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
- int **smi_ospf6_if_transmit_delay_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t transmitDelay, int instanceId)
- int **smi_ospf6_if_transmit_delay_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
- int **smi_ospf6_if_retransmit_interval_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t retransmitInterval, int instanceId)
- int **smi_ospf6_if_retransmit_interval_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)

- int **smi_ospf6_if_cost_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t ospf6IfCost, int instanceId)
- int **smi_ospf6_if_cost_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
- int **smi_ospf6_if_priority_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int ospf6IfPriority, int instanceId)
- int **smi_ospf6_if_priority_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
- int **smi_ospf6_if_disable_all_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int **smi_ospf6_if_disable_all_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName)
- int **smi_ospf6_if_disable_all_set_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int isDisable)
- int **smi_ospf6_if_disable_all_set_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int isDisable)
- int **smi_ospf6_if_link_lsa_suppression_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int ospf6LsaSuppressionValue, int instanceId)
- int **smi_ospf6_if_te_metric_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, u_int32_t ospf6TeMetric, int instanceId)
- int **smi_ospf6_if_te_metric_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId)
- int **smi_ospf6_restart_graceful_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, u_int16_t gracePeriod, int ospf6RestartGracefulReason)
- int **smi_ospf6_restart_graceful_set_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, u_int16_t gracePeriod)
- int **smi_ospf6_restart_graceful_set_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, u_int16_t gracePeriod)
- int **smi_ospf6_graceful_restart_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, int gracePeriod, int ospf6RestartGracefulReason)
- int **smi_ospf6_graceful_restart_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int **smi_ospf6_restart_helper_policy_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, int ospf6HelperPolicy)
- int **smi_ospf6_hitless_restart_helper_never_policy_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, int hitlessRestartHelpNever)
- int **smi_ospf6_hitless_restart_helper_never_policy_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, int hitlessRestartHelpNever)
- int **smi_ospf6_hitless_restart_helper_policy_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, int ospf6HitlessHelperPolicy)
- int **smi_ospf6_restart_helper_policy_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, int ospf6HelperPolicy)
- int **smi_ospf6_restart_helper_grace_period_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, int gracePeriod)
- int **smi_ospf6_restart_helper_never_router_id_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, struct pal_in4_addr routerId)

- int **smi_ospf6_restart_helper_never_router_id_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, struct pal_in4_addr routerId)
- int **smi_ospf6_restart_helper_never_router_unset_all_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int **smi_ospf6_restart_helper_grace_period_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int **smi_ospf6_restart_helper_policy_unset_all_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int **smi_ospf6_capability_restart_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6CapabilityRestartMethod)
- int **smi_ospf6_capability_restart_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_ipv6_ospf_display_route_single_line_set_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int **smi_ospf6_ipv6_ospf_display_route_single_line_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId)
- int **smi_ospf6_router_id_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr routerId)
- int **smi_ospf6_router_id_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_address_family_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_char addFamily)
- int **smi_ospf6_address_family_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_char addFamily)
- int **smi_ospf6_address_family_set_sdkapi** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, bool_t add_family)
- int **smi_ospf6_address_family_unset_sdkapi** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, bool_t add_family)
- int **smi_ospf6_address_family_unset_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, bool_t add_family)
- int **smi_ospf6_address_family_set_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, bool_t add_family)
- int **smi_ospf6_abr_type_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int abrType)
- int **smi_ospf6_abr_type_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_timers_spf_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t minDelay, u_int32_t maxDelay)
- int **smi_ospf6_timers_spf_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_auto_cost_reference_bandwidth_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t ospf6ReferenceBandWidth)
- int **smi_ospf6_auto_cost_reference_bandwidth_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_max_concurrent_dd_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int16_t maxDBDescriptor)

- int **smi_ospf6_max_concurrent_dd_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_max_unuse_packet_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t maxUnusePkt)
- int **smi_ospf6_max_unuse_packet_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_max_unuse_lsa_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t maxUnuseLsa)
- int **smi_ospf6_max_unuse_lsa_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_area_format_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat)
- int **smi_ospf6_area_stub_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat)
- int **smi_ospf6_area_stub_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- int **smi_ospf6_area_no_summary_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- int **smi_ospf6_area_default_cost_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6DefaultCost, int ospf6AreaFormat)
- int **smi_ospf6_area_default_cost_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- int **smi_ospf6_area_range_ipv4_set_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat, struct pal_in4_addr ipv4Addr, int prefixlen, int ospf6AreaRangeStatus)
- int **smi_ospf6_area_range_ipv4_unset_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr ipv4Addr, int prefixlen)
- int **smi_ospf6_area_nssa_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat)
- int **smi_ospf6_area_nssa_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- int **smi_ospf6_area_nssa_translator_role_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, u_char ospf6AreaNssaTransRole)
- int **smi_ospf6_area_nssa_translator_role_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- int **smi_ospf6_area_nssa_default_originate_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- int **smi_ospf6_area_nssa_default_originate_metric_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6DefaultOriginMetric)

- **int smi_ospf6_area_nssa_default_originate_metric_type_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6DefaultOriginMetricType)
- **int smi_ospf6_area_nssa_default_originate_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- **int smi_ospf6_area_nssa_no_redistribution_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- **int smi_ospf6_area_nssa_no_redistribution_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId)
- **int smi_ospf6_area_nssa_stability_interval_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, u_int32_t stabilityInterval)
- **int smi_ospf6_vlink_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int ospf6AreaFormat)
- **int smi_ospf6_vlink_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)
- **int smi_ospf6_vlink_dead_interval_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkDeadInterval)
- **int smi_ospf6_vlink_dead_interval_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)
- **int smi_ospf6_vlink_hello_interval_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkHelloInterval)
- **int smi_ospf6_vlink_hello_interval_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)
- **int smi_ospf6_vlink_retransmit_interval_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkRetransmitInterval)
- **int smi_ospf6_vlink_retransmit_interval_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)
- **int smi_ospf6_vlink_transmit_delay_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkTransmitDelay)
- **int smi_ospf6_vlink_transmit_delay_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)
- **int smi_ospf6_vlink_instance_id_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int VirtualLinkInstanceId)

- int **smi_ospf6_vlink_instance_id_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)
- int **smi_ospf6_capability_cspf_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_capability_cspf_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_enable_db_summary_opt_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_disable_db_summary_opt_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_redistribute_set_by_ospf_src_tag_sdkapi_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *ospf6RedistProtocol, int sourceOspfTag)
- int **smi_ospf6_redistribute_set_by_ospf_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)
- int **smi_ospf6_redistribute_set_by_ospf6_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag)
- int **smi_ospf6_redistribute_unset_by_ospf_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag)
- int **smi_ospf6_redistribute_unset_by_ospf6_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)
- int **smi_ospf6_redistribute_metric_type_set_by_ospf_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag, u_char ospf6RedistMetricType)
- int **smi_ospf6_redistribute_metric_type_set_by_ospf6_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag, u_char ospf6RedistMetricType)
- int **smi_ospf6_redist_default_metric_type_set_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_char ospf6RedistDefaultMetric)
- int **smi_ospf6_redist_default_metric_type_set_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_char ospf6RedistDefaultMetric)
- int **smi_ospf6_redist_default_metric_set_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t ospf6RedistDefaultMetricValue)
- int **smi_ospf6_redist_default_metric_set_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t ospf6RedistDefaultMetricValue)
- int **smi_ospf6_redist_default_metric_type_unset_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_redist_default_metric_type_unset_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)

- int **smi_ospf6_redist_default_metric_unset_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_redist_default_metric_unset_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_redistribute_metric_type_unset_by_ospf_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)
- int **smi_ospf6_redistribute_metric_type_unset_by_ospf6_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag)
- int **smi_ospf6_redistribute_metric_set_by_ospf_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag, u_int32_t ospf6RedistMetricValue)
- int **smi_ospf6_redistribute_metric_set_by_ospf6_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag, u_int32_t ospf6RedistMetricValue)
- int **smi_ospf6_redistribute_metric_unset_by_ospf_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)
- int **smi_ospf6_redistribute_metric_unset_by_ospf6_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag)
- int **smi_ospf6_redistribute_route_tag_unset_by_ospf_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)
- int **smi_ospf6_redistribute_route_tag_unset_by_ospf6_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag)
- int **smi_ospf6_redistribute_route_tag_set_by_ospf6_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag, u_int32_t ospf6RedistTagVal)
- int **smi_ospf6_redistribute_route_tag_set_by_ospf_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag, u_int32_t ospf6RedistTagVal)
- int **smi_ospf6_default_metric_set_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int32_t ospf6DefaultMetric)
- int **smi_ospf6_default_metric_unset_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag)
- int **smi_ospf6_routemap_set_by_ospf_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag, char *routeMapName)
- int **smi_ospf6_routemap_set_by_ospf6_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspfTag, char *routeMapName)
- int **smi_ospf6_routemap_unset_by_ospf_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)

- int **smi_ospf6_routemap_unset_by_ospf6_src_tag_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int ospf6RedistProtocol, char *sourceOspf6Tag)
- int **smi_ospf6_routemap_default_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, char *routeMapName)
- int **smi_ospf6_routemap_default_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag)
- int **smi_ospf6_redist_default_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int ospf6RedistDefaultOrigin)
- int **smi_ospf6_redist_default_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag)
- int **smi_ospf6_router_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag)
- int **smi_ospf6_router_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag)
- int **smi_ospf6_passive_if_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ifName, char *ospf6ProcessTag)
- int **smi_ospf6_passive_if_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ifName, char *ospf6ProcessTag)
- int **smi_ospf6_passive_if_default_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag)
- int **smi_ospf6_passive_if_default_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag)
- int **smi_ospf6_distance_all_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int ospf6AdminDistance)
- int **smi_ospf6_distance_all_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag)
- int **smi_ospf6_distance_intra_area_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int intraDistance)
- int **smi_ospf6_distance_intra_area_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag)
- int **smi_ospf6_distance_inter_area_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int interDistance)
- int **smi_ospf6_distance_inter_area_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag)
- int **smi_ospf6_distance_external_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int externalDistance)
- int **smi_ospf6_distance_external_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag)
- int **smi_ospf6_distribute_list_in_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, char *accessListName)
- int **smi_ospf6_distribute_list_in_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, char *accessListName)
- int **smi_ospf6_distribute_list_out_set_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int ospf6RedistProtocol, char *ospf6DistributeTag, char *accessListName)
- int **smi_ospf6_distribute_list_out_unset_validate** (struct smiclient_globals *azg, u_int32_t vrid, char *ospf6ProcessTag, int ospf6RedistProtocol, char *stag, char *accessListName)

- int **smi_ospf6_summary_address_ipv4_set_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)
- int **smi_ospf6_summary_address_ipv4_unset_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)
- int **smi_ospf6_summary_address_tag_ipv4_set_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen, u_int32_t routeTag)
- int **smi_ospf6_summary_address_tag_ipv4_unset_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)
- int **smi_ospf6_summary_address_not_advertise_ipv4_set_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)
- int **smi_ospf6_summary_address_not_advertise_ipv4_unset_sdkapi_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)
- int **smi_ospf6_set_area_nssa_trans_role_validate** (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, u_int32_t ospf6AreaNssaTransRole)
- int **smi_ospf6_set_area_stub_metric_type_validate** (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6StubMetricType)
- int **smi_ospf6_set_area_te_enabled_validate** (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6TeStatus)
- int **smi_ospf6_set_if_type_validate** (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int ospf6IfType)
- int **smi_ospf6_set_if_admin_stat_validate** (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int adminstat)
- int **smi_ospf6_set_if_rtr_dead_interval_validate** (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int ifRtrDeadInterval)
- int **smi_ospf6_show_topology** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, struct pal_in4_addr areaId, int startIndex, int endIndex, struct list *topologyData, int(*funpointer)(struct smiTopologyList *getinfo))
- int **smi_ospf6_set_asbdr_rtr_status_validate** (struct smiclient_globals *azg, char *ospf6ProcessTag, int ospf6AsbdrRtrStatus)
- int **smi_ospf6_debug_validate** (struct smiclient_globals *azg, u_int32_t vrId, int ospf6Debug)
- int **smi_ospf6_no_debug_validate** (struct smiclient_globals *azg, u_int32_t vrId, int ospf6Debug)
- int **smi_debug_ospf6_packet_validate** (struct smiclient_globals *azg, int vrId, int packetType, int ospf6DebugMode, int ospf6DebugDetail)
- int **smi_debug_no_ospf6_packet_validate** (struct smiclient_globals *azg, int vrId, int packetType, int ospf6DebugMode, int ospf6DebugDetail)
- int **smi_ospf6_set_area_te_enabled_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, int ospf6TeStatus)

- int **smi_ospf6_set_area_te_enabled_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, int ospf6TeStatus)
- int **smi_ospf6_area_summary_set_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, bool_t isAreaStub, int ospf6AreaSummary)
- int **smi_ospf6_area_stub_unset_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, bool_t isAreaStub)
- int **smi_ospf6_area_stub_unset_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, bool_t isAreaStub)
- int **smi_ospf6_area_summary_set_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, bool_t isAreaStub, int ospf6AreaSummary)
- int **smi_ospf6_area_stub_set_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, bool_t isAreaStub)
- int **smi_ospf6_area_stub_set_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, bool_t isAreaStub)
- int **smi_ospf6_area_nssa_summary_set_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, bool_t ospf6Nssa, int nssaAreaSummary)
- int **smi_ospf6_area_nssa_summary_set_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, bool_t ospf6Nssa, int nssaAreaSummary)
- int **smi_ospf6_set_area_stub_metric_type_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, int ospf6StubMetricType)
- int **smi_ospf6_set_area_stub_metric_type_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, char *areaId, int ospf6StubMetricType)
- int **smi_ospf6_if_ipv6_router_set_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, char *areaId, char *ospf6ProcessTag, int instanceId)
- int **smi_ospf6_if_ipv6_router_set_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, char *areaId, char *ospf6ProcessTag, int instanceId)
- int **smi_ospf6_if_ipv6_router_unset_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, char *areaId, char *ospf6ProcessTag, int instanceId)
- int **smi_ospf6_if_ipv6_router_no_process_set_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, char *areaId, bool_t ifProcessTagPresent, int processInstanceId)
- int **smi_ospf6_if_ipv6_router_no_process_set_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, char *areaId, bool_t ifProcessTagPresent, int processInstanceId)
- int **smi_ospf6_if_ipv6_router_no_process_unset_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vr_id, char *name, char *area_id, bool_t ifProcessTagPresent, int instance_id)
- int **smi_ospf6_if_ipv6_router_no_process_unset_wrapper** (struct smiclient_globals *azg, u_int32_t vr_id, char *name, char *area_id, bool_t ifProcessTagPresent, int instance_id)

- **int smi_ospf6_if_mtu_ignore_set_wrapper** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId, bool_t ifMtuIgnore)
- **int smi_ospf6_if_mtu_ignore_set_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ifName, int instanceId, bool_t ifMtuIgnore)
- **int smi_ospf6_debug_wrapper** (struct smiclient_globals *azg, u_int32_t vr_id, int ospf6Debug, bool_t isDebug)
- **int smi_debug_ospf6_packet_wrapper** (struct smiclient_globals *azg, int vr_id, int ospf6Packet, bool_t ospf6PacketDebug)
- **int smi_ospf6_debug_wrapper_validate** (struct smiclient_globals *azg, u_int32_t vr_id, int ospf6Debug, bool_t isDebug)
- **int smi_debug_ospf6_packet_wrapper_validate** (struct smiclient_globals *azg, int vr_id, int ospf6Packet, bool_t ospf6PacketDebug)

2.1.1 Detailed Description

Provides API for managing OSPF. The API provided in this file forms the basis of ZebOS OSPFv3 management. These APIs are used by various north bound management interfaces like CLI, SNMP and SMI

2.1.2 Function Documentation

2.1.2.1 int merge_ospf6_list_nbr_info (struct list * *listDest*, struct sminbrByTagList * *listSrc*)

This function returns the list of neighbor by tag. smi_ospf6_show_neighbor_by_tag

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrId** Virtual router ID; for a non-virtual-router implementation, specify 0
- ← **tag** OSPF process tag
- ← **startIndex** Starting index of the node of the returned list, if not specified it is considered be 0
- ← **endIndex** Ending index of the node of the returned list, if not specified it is considered be 0
- **nbrList** Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.2 int smi_debug_no_ospf6_packet (struct smiclient_globals * *azg*, int *vrId*, int *packetType*, int *ospf6DebugMode*, int *ospf6DebugDetail*)

use this function to disable the packet debugging options for ospfv3 zebos information
smi_debug_no_ospf6_packet

Parameters:

- ← *azg* pointer to the smi client global structure
- ← *vrId* virtual router id
- ← *packetType* pass packet type as folowing:
 - smi_ospf6_dbg_packet_all - debug all type of ospf packets
 - smi_ospf6_dbg_packet_hello - debug ospf hello packets
 - smi_ospf6_dbg_packet_desc - debug ospf database description
 - smi_ospf6_dbg_packet_req - debug ospf link state request
 - smi_ospf6_dbg_packet_upd - debug ospf link state update
 - smi_ospf6_dbg_packet_ack - debug ospf link state acknowledgment
- ← *debugMode* pass debug mode as folowing:
 - smi_ospf6_dbg_packet - debug both sent and recieved packets
 - smi_ospf6_dbg_packet_send - debug sent packets
 - smi_ospf6_dbg_packet_rcv - debug recieved packets
- ← *isDetail* pass is detail flag as folowing:
 - pal_true - debug detail information
 - pal_false

Returns:

- 0 on success, otherwise one of the following error codes `ospf_api_set_err_vr_`-
`not_exist`
`smi_error`

2.1.2.3 `int smi_debug_ospf6_packet (struct smiclient_globals * azg, int vrId, int packetType, int ospf6DebugMode, int ospf6DebugDetail)`

Use this function to specify the packet debugging options for OSPFv3 ZebOS information. `smi_debug_ospf6_packet`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *packetType* Pass packet type as folowing:
 - SMI_OSPF6_DBG_PACKET_ALL - Debug all type of OSPF packets
 - SMI_OSPF6_DBG_PACKET_HELLO - Debug OSPF hello packets
 - SMI_OSPF6_DBG_PACKET_DESC - Debug OSPF database description
 - SMI_OSPF6_DBG_PACKET_REQ - Debug OSPF link state request
 - SMI_OSPF6_DBG_PACKET_UPD - Debug OSPF link state update
 - SMI_OSPF6_DBG_PACKET_ACK - Debug OSPF link state acknowledgment

- ← *debugMode* Pass debug mode as following:
 - SMI_OSPF6_DBG_PACKET - Debug both sent and recieved packets
 - SMI_OSPF6_DBG_PACKET_SEND - Debug sent packets
 - SMI_OSPF6_DBG_PACKET_RCV - Debug recieved packets
- ← *isDetail* Pass is detail flag as following:
 - PAL_TRUE - Debug detail information
 - PAL_FALSE

Returns:

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

2.1.2.4 **int smi_ospf6_abr_type_set** (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ospf6ProcessTag*, int *abrType*)

This function sets area border router(ABR) type. smi_ospf6_abr_type_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *abrType* The type of area border router <1-3>
 - 1-OSPF_ABR_TYPE_STANDARD
 - 2-OSPF_ABR_TYPE_CISCO
 - 3-OSPF_ABR_TYPE_IBM

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
OSPF6_API_SET_ERR_ABR_TYPE_INVALID

2.1.2.5 **int smi_ospf6_abr_type_unset** (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ospf6ProcessTag*)

This function sets area border router(ABR) type to OSPF_ABR_TYPE_CISCO. smi_ospf6_abr_type_unset

Parameters:

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

← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

**2.1.2.6 int smi_ospf6_address_family_set (struct smiclient_globals * *azg*,
 u_int32_t *vrId*, char * *ospf6ProcessTag*, u_char *addFamily*)**

This function sets family mode to IPv4 unicast or IPv6 unicast. smi_ospf6_address_family_set

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *addFamily* Address family
 0-AF_IPV6_UNICAST
 2AF_IPV4_UNICAST

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

**2.1.2.7 int smi_ospf6_address_family_unset (struct smiclient_globals * *azg*,
 u_int32_t *vrId*, char * *ospf6ProcessTag*, u_char *addFamily*)**

This function sets family mode to IPv4 unicast or IPv6 unicast. smi_ospf6_address_family_unset

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *addFamily* Address family
 0-AF_IPV6_UNICAST
 2AF_IPV4_UNICAST

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.8 `int smi_ospf6_area_default_cost_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6DefaultCost, int ospf6AreaFormat)`

This function configures the ospf6DefaultCost for default summary route sent into a stub area. smi_ospf6_area_default_cost_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *ospf6DefaultCost* Stub advertised numeric <0-16777215>
- ← *format* Area ID format <0-2>
 - 0-OSPF_areaId_FORMAT_DEFAULT,
 - 1-OSPF_areaId_FORMAT_ADDRESS,
 - 2-OSPF_areaId_FORMAT_DECIMAL

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_IS_BACKBONE
- OSPF6_API_SET_ERR_METRIC_INVALID
- OSPF6_API_SET_ERR_AREA_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_IS_DEFAULT

2.1.2.9 `int smi_ospf6_area_default_cost_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId)`

This function removes the assigned stub default ospf6DefaultCost. smi_ospf6_area_default_cost_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_IS_DEFAULT

2.1.2.10 `int smi_ospf6_area_format_set (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId,
int ospf6AreaFormat)`

This function sets the format of area ID. smi_ospf6_area_format_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *areaFormat* Area ID format <0-2>
 - 0-OSPF_AREA_ID_FORMAT_DEFAULT,
 - 1-OSPF_AREA_ID_FORMAT_ADDRESS,
 - 2-OSPF_AREA_ID_FORMAT_DECIMAL

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_NOT_EXIST
- OSPF6_API_SET_ERR_areaId_FORMAT_INVALID

2.1.2.11 `int smi_ospf6_area_no_summary_set (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId)`

This function configures an area as stub no-summary: inter-area routes are not injected into stub. smi_ospf6_area_no_summary_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_IS_DEFAULT

2.1.2.12 `int smi_ospf6_area_nssa_default_originate_metric_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6DefaultOriginMetric)`

This function originates Type-7 default LSAs with a specific metric into an NSSA area. `smi_ospf6_area_nssa_default_originate_metric_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *metric* Metric value for default routes numeric <0-16777214>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_METRIC_INVALID
 OSPF6_API_SET_ERR_AREA_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_NOT_NSSA

2.1.2.13 `int smi_ospf6_area_nssa_default_originate_metric_type_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6DefaultOriginMetricType)`

This function originates Type-7 default LSAs with a specific metric type into an NSSA area. `smi_ospf6_area_nssa_default_originate_metric_type_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *metricType* Metric type numeric <1-2>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_METRIC_TYPE_INVALID
 OSPF6_API_SET_ERR_AREA_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_NOT_NSSA

2.1.2.14 `int smi_ospf6_area_nssa_default_originate_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId)`

This function originates Type-7 default LSAs into an NSSA area. `smi_ospf6_area_nssa_default_originate_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_AREA_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_NOT_NSSA

2.1.2.15 `int smi_ospf6_area_nssa_default_originate_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId)`

This function stops originating Type-7 default LSAs into an NSSA area. `smi_ospf6_area_nssa_default_originate_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_AREA_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_NOT_NSSA

2.1.2.16 `int smi_ospf6_area_nssa_no_redistribution_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId)`

This function redistributes routes into an NSSA area. `smi_ospf6_area_nssa_no_redistribution_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_AREA_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_NOT_NSSA

2.1.2.17 `int smi_ospf6_area_nssa_no_redistribution_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId)`

This function stops redistributing routes into an NSSA area. `smi_ospf6_area_nssa_no_redistribution_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_AREA_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_NOT_NSSA

2.1.2.18 `int smi_ospf6_area_nssa_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat)`

This function sets an area as an NSSA (Not-So-Stubby-Area). `smi_ospf6_area_nssa_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
OSPF6_API_SET_ERR_AREA_IS_BACKBONE
OSPF6_API_SET_ERR_AREA_HAS_VLINK
OSPF6_API_SET_ERR_AREA_IS_STUB

2.1.2.19 `int smi_ospf6_area_nssa_stability_interval_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, u_int32_t stabilityInterval)`

This function sets the stability interval for an NSSA area. `smi_ospf6_area_nssa_stability_interval_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *stabilityInterval* Stability interval in seconds <0-2147483647>

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
OSPF6_API_SET_ERR_AREA_IS_BACKBONE
OSPF6_API_SET_ERR_AREA_NOT_EXIST
OSPF6_API_SET_ERR_AREA_NOT_NSSA

2.1.2.20 `int smi_ospf6_area_nssa_translator_role_set (struct smiclient_globals * azg, u_int32_t vrid, char * ospf6ProcessTag, struct pal_in4_addr areaId, u_char ospf6AreaNssaTransRole)`

This function sets translator role for an NSSA (Not-So-Stubby-Area). `smi_ospf6_area_nssa_translator_role_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrid* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *role* NSSA-ABR translator role <1-2>
 - 1-OSPF_NSSA_TRANSLATE_ALWAYS
 - 2-OSPF_NSSA_TRANSLATE_CANDIDATE

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_AREA_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_NOT_NSSA

2.1.2.21 `int smi_ospf6_area_nssa_translator_role_unset (struct smiclient_globals * azg, u_int32_t vrid, char * ospf6ProcessTag, struct pal_in4_addr areaId)`

This function removes translator role for an NSSA (Not-So-Stubby-Area). `smi_ospf6_area_nssa_translator_role_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrid* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_AREA_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_NOT_NSSA

2.1.2.22 `int smi_ospf6_area_nssa_unset (struct smiclient_globals * azg, u_int32_t vrid, char * ospf6ProcessTag, struct pal_in4_addr areaId)`

This function unsets an area as an NSSA (Not-So-Stubby-Area). `smi_ospf6_area_nssa_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrid* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tagl
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_IS_BACKBONE
- OSPF6_API_SET_ERR_AREA_IS_STUB

2.1.2.23 `int smi_ospf6_area_range_ipv4_set_sdkapi (struct smiclient_globals * azg, u_int32_t vrid, char * ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat, struct pal_in4_addr ipv4Addr, int prefixlen, int ospf6AreaRangeStatus)`

This function configures an OSPF address range. `smi_ospf6_area_range_ipv4_set_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrid* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *ipv4Addr* IPv4 address
- ← *prefixlen* Area range IPv4 prefix length numeric <0-128>
- ← *status* True/False whether range has to be advertised
 - Advertise this range-PAL_TRUE
 - Donot advertise this range-PAL_FALSE

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_PREFIXLEN_INVALID

`smi_ospf6_area_range_ipv4_set_sdkapi`

This function configures an OSPF address range.

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *ipv4Addr* IPv4 address
- ← *prefixlen* Area range IPv4
- ← *status* True/False whether range has to be advertised
 Advertise this range-PAL_TRUE
 Donot advertise this range-PAL_FALSE

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_PREFIXLEN_INVALID

2.1.2.24 `int smi_ospf6_area_range_ipv4_unset_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr ipv4Addr, int prefixlen)`

This function removes the configured area range. `smi_ospf6_area_range_ipv6_unset_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *ipv6Addr* IPv6 address
- ← *prefixlen* Area range IPv6 prefix length numeric <0-128>

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_PREFIXLEN_INVALID
- OSPF6_API_SET_ERR_AREA_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_RANGE_NOT_EXIST

smi_ospf6_area_range_ipv4_unset_sdkapi

This function removes the configured area range.

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *ipv4Addr* IPv4 starting address.
- ← *prefixlen* prefix length

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_PREFIXLEN_INVALID
- OSPF6_API_SET_ERR_AREA_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_RANGE_NOT_EXIST

2.1.2.25 `int smi_ospf6_area_stub_no_summary_set (struct smiclient_globals *
azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr
areaId, int ospf6AreaFormat)`

This function makes an area a normal area. smi_ospf6_area_stub_no_summary_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *format* Area ID format <0-2>
 - 0-OSPF_areaId_FORMAT_DEFAULT,
 - 1-OSPF_areaId_FORMAT_ADDRESS,
 - 2-OSPF_areaId_FORMAT_DECIMAL

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_AREA_IS_BACKBONE
- OSPF6_API_SET_ERR_AREA_HAS_VLINK
- OSPF6_API_SET_ERR_AREA_IS_NSSA

2.1.2.26 `int smi_ospf6_area_stub_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, int ospf6AreaFormat)`

This function makes an area a normal area. `smi_ospf6_area_stub_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_AREA_HAS_VLINK
 OSPF6_API_SET_ERR_AREA_IS_NSSA

2.1.2.27 `int smi_ospf6_area_stub_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId)`

This function unsets area as stub configuration. `smi_ospf6_area_stub_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_AREA_IS_NSSA

2.1.2.28 `int smi_ospf6_auto_cost_reference_bandwidth_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, u_int32_t ospf6ReferenceBandWidth)`

This function sets the reference bandwidth value. OSPF calculates the OSPF metric for an interface by dividing the reference bandwidth. `smi_ospf6_auto_cost_reference_bandwidth_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *refbw* The reference bandwidth in Mbits/sec numeric <1-4294967>

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
OSPF6_API_SET_ERR_REFERENCE_BANDWIDTH_INVALID

2.1.2.29 int smi_ospf6_auto_cost_reference_bandwidth_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)

This function sets the reference bandwidth to its default value (100000 kbps). smi_ospf6_auto_cost_reference_bandwidth_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.30 int smi_ospf6_capability_cspf_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)

This function enables the CSPF capability for an OSPFv3 process. smi_ospf6_capability_cspf_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes

```

OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
OSPF6_API_SET_ERR_CSPF_INSTANCE_EXIST
OSPF6_API_SET_ERR_CSPF_CANT_START

```

2.1.2.31 `int smi_ospf6_capability_cspf_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)`

This function disables the CSPF capability for an OSPFv3 process. `smi_ospf6_capability_cspf_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_CSPF_INSTANCE_NOT_EXIST

2.1.2.32 `int smi_ospf6_capability_restart_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6CapabilityRestartMethod)`

This function enables OSPFv3 graceful restart capability. `smi_ospf6_capability_restart_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 process tag
- ← *method* Graceful restart

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.33 int smi_ospf6_capability_restart_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ospf6ProcessTag*)

This function disables OSPFv3 graceful restart capability. smi_ospf6_capability_restart_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.34 int smi_ospf6_debug (struct smiclient_globals * *azg*, u_int32_t *vrId*, int *ospf6Debug*)

Use this function to specify debugging option for OSPFv3 ZebOS information. smi_ospf6_debug

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *debug* Pass debug flag as following:
 - SMI_OSPF6_DBG_ALL - Enable or disable debugging for ifsm,nsfm,lsa,nsm,events and route
 - SMI_OSPF6_DBG_IFSM - Debug OSPF Interface State Machine
 - SMI_OSPF6_DBG_IFSM_EVENTS - Debug OSPF Interface State Machine events information
 - SMI_OSPF6_DBG_IFSM_STATUS - Debug OSPF Interface State Machine status information
 - SMI_OSPF6_DBG_IFSM_TIMERS - Debug OSPF Interface State Machine timers information
 - SMI_OSPF6_DBG_NFSM - Debug OSPF Neighbor State Machine
 - SMI_OSPF6_DBG_NFSM_EVENTS - Debug OSPF Neighbor State Machine events information
 - SMI_OSPF6_DBG_NFSM_STATUS - Debug OSPF Neighbor State Machine status information
 - SMI_OSPF6_DBG_NFSM_TIMERS - Debug OSPF Neighbor State Machine timers information
 - SMI_OSPF6_DBG_LSA - Debug OSPF Link State Advertisement
 - SMI_OSPF6_DBG_LSA_FLOODING - Debug LSA flooding

SMI_OSPF6_DBG_LSA_GENERATE - Debug LSA generation
 SMI_OSPF6_DBG_LSA_INSTALL - Debug LSA installation
 SMI_OSPF6_DBG_LSA_MAXAGE - Debug the maximum age processing
 SMI_OSPF6_DBG_LSA_REFRESH - Debug LSA refresh
 SMI_OSPF6_DBG_NSM - Debug OSPF NSM information
 SMI_OSPF6_DBG_NSM_INTERFACE - Debug NSM interface information
 SMI_OSPF6_DBG_NSM_REDISTRIBUTE - Debug NSM redistribute information
 SMI_OSPF6_DBG_RIB - Debug OSPF RIB information
 SMI_OSPF6_DBG_RIB_INTERFACE - Debug RIB interface information
 SMI_OSPF6_DBG_RIB_REDISTRIBUTE - Debug RIB redistribute information
 SMI_OSPF6_DBG_EVENT - Debug OSPF event trouble shooting
 SMI_OSPF6_DBG_EVENT_ABR - Debug OSPF ABR events
 SMI_OSPF6_DBG_EVENT_ASBR - Debug OSPF ASBR events
 SMI_OSPF6_DBG_EVENT_LSA - Debug OSPF LSA events
 SMI_OSPF6_DBG_EVENT_NSSA - Debug OSPF NSSA events
 SMI_OSPF6_DBG_EVENT_OS - Debug OSPF OS events
 SMI_OSPF6_DBG_EVENT_ROUTER - Debug OSPF ROUTER events
 SMI_OSPF6_DBG_EVENT_VLINK - Debug OSPF VLINK events
 SMI_OSPF6_DBG_ROUTE_CALC - Debug route calculation
 SMI_OSPF6_DBG_ROUTE_CALC_ASE - Debug OSPF external route calculation
 SMI_OSPF6_DBG_ROUTE_CALC_IA - Debug OSPF inter-area route calculation
 SMI_OSPF6_DBG_ROUTE_INSTALL - Debug OSPF route installation
 SMI_OSPF6_DBG_ROUTE_CALC_SPF - Debug OSPF SPF calculation
 SMI_OSPF6_DBG_BFD - Debug Bidirectional Forwarding Detection

Returns:

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

2.1.2.35 int smi_ospf6_default_metric_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, u_int32_t ospf6DefaultMetric)

This function sets default metric values for the OSPFv3 routing protocol. smi_ospf6_default_metric_set

Parameters:

← *azg* Pointer to the SMI client global structure

- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *metric* Default metric <0-16777214>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_METRIC_INVALID

2.1.2.36 int smi_ospf6_default_metric_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)

This function unsets configured default metric. smi_ospf6_default_metric_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.37 int smi_ospf6_disable_db_summary_opt (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)

This function disables the OSPFv3 Database Summary List optimization. smi_ospf6_disable_db_summary_opt

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.38 `int smi_ospf6_distance_all_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6AdminDistance)`

This function sets OSPFv3 route administrative distances based on route type. It sets distance for an entire group of routes rather than a specific route that passes an access list. `smi_ospf6_distance_all_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *distance* Route distance numeric <1-254>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_DISTANCE_INVALID

2.1.2.39 `int smi_ospf6_distance_all_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)`

This function unsets OSPFv3 route administrative distances based on route type. `smi_ospf6_distance_all_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.40 `int smi_ospf6_distance_external_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int externalDistance)`

This function sets distance for external routes. `smi_ospf6_distance_external_set`

Parameters:

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

- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *distance* Distance value numeric <1-254>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_DISTANCE_INVALID

2.1.2.41 int smi_ospf6_distance_external_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ospf6ProcessTag*)

This function unsets distance for external routes. smi_ospf6_distance_external_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.42 int smi_ospf6_distance_inter_area_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ospf6ProcessTag*, int *interDistance*)

This function sets distance for inter-area routes. smi_ospf6_distance_inter_area_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *distance* Distance value numeric <1-254>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_DISTANCE_INVALID

2.1.2.43 `int smi_ospf6_distance_inter_area_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)`

This function unsets distance for inter-area routes. `smi_ospf6_distance_inter_area_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.44 `int smi_ospf6_distance_intra_area_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int intraDistance)`

This function sets distance for intra-area routes. `smi_ospf6_distance_intra_area_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *distance* Distance value numeric <1-254>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_DISTANCE_INVALID

2.1.2.45 `int smi_ospf6_distance_intra_area_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)`

This function unsets distance for intra-area routes. `smi_ospf6_distance_intra_area_unset`

Parameters:

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

← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.46 int smi_ospf6_distribute_list_in_set (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ospf6ProcessTag*, char * *accessListName*)

This function sets access-list to filter networks for incoming routing updates. It redistributes other routing protocols into the OSPF routing table. smi_ospf6_distribute_list_in_set

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *accessListName* Access-list name

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_MALLOC_ERR

2.1.2.47 int smi_ospf6_distribute_list_in_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, char * *ospf6ProcessTag*, char * *accessListName*)

This function unsets access-list to filter networks for incoming routing updates. smi_ospf6_distribute_list_in_unset

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *accessListName* Access-list name

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.48 `int smi_ospf6_distribute_list_out_set (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistProtocol, char
* ospf6DistributeTag, char * accessListName)`

This function sets access-list and redistributed protocol type filter networks for outgoing routing updates. It redistributes other routing protocols into the OSPF routing table. `smi_ospf6_distribute_list_out_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *stag* OSPFv3 process tag or OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance
- ← *accessListName* Access-list name

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_SELF_TAG
 OSPF6_API_SET_MALLOC_ERR

2.1.2.49 `int smi_ospf6_distribute_list_out_unset (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistProtocol, char
* stag, char * accessListName)`

This function unsets access-list and redistributed protocol type filter networks for outgoing routing updates. `smi_ospf6_distribute_list_out_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

- ← **proto** Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← **tag** OSPFv3 process tag or OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance
- ← **accessListName** Access-list name

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_SELF_TAG
 OSPF6_API_SET_MALLOC_ERR

2.1.2.50 int smi_ospf6_enable_db_summary_opt (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)

This function enables the OSPFv3 Database Summary List optimization. smi_ospf6_enable_db_summary_opt

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrId** Virtual Router ID numeric <0-255>
- ← **ospf6ProcessTag** OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.51 int smi_ospf6_get_admin_stat (struct smiclient_globals * azg, char * ospf6ProcessTag, int * adminStat)

This function gets administrative status of OSPFv3 in the router. The value 'enabled' denotes that the OSPFv3 Process is active on at least one. smi_ospf6_get_admin_stat

Parameters:

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

← **tag** OSPFv3 Process tag
 → **adminStat** OSPF administration status <1-2>
 1-OSPF6_API_STATUS_ENABLED
 2-OSPF6_API_STATUS_DISABLED

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.52 `int smi_ospf6_get_area_bdr_rtr_count (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int * areaBdrRtrCount)`

This function gets total number of area border routers reachable within this area. This is initially zero, and is calculated in each Shortest Path First (SPF) pass. `smi_ospf6_get_area_bdr_rtr_count`

Parameters:

← **azg** Pointer to the SMI client global structure
 ← **tag** OSPFv3 Process tag
 ← **areaId** OSPFv3 area ID in an IPv4 address format
 → **areaBdrRtrCount** Total no of area border routers reachable

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.53 `int smi_ospf6_get_area_bdr_rtr_status (struct smiclient_globals * azg, char * ospf6ProcessTag, int * areaBdrRtrStatus)`

This function gets the flag to denote whether this router is an area border router. `smi_ospf6_get_area_bdr_rtr_status`

Parameters:

← **azg** Pointer to the SMI client global structure
 ← **tag** OSPFv3 Process tag
 → **areaBdrRtrStatus** Flag value (OSPF6_API_TRUE|OSPF6_API_FALSE)

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.54 `int smi_ospf6_get_area_lsa_cksum_sum (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int * areaLsaCksumSum)`

This function gets the 32-bit unsigned sum of the Area-scope link state advertisements' LS checksums contained in this area's link state database. The sum can be used to determine if there has been a change in a router's link state database or to compare the link state database of two routers. `smi_ospf6_get_area_lsa_cksum_sum`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- *areaLsaCksumSum* Sum of LS checksums contained in this area's link state database

Returns:

- 0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.55 `int smi_ospf6_get_area_lsa_count (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int * areaLsaCount)`

This function gets the total number of Area-scope link state advertisements in this area's link state database. `smi_ospf6_get_area_lsa_count`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- *areaLsaCount* Total number of Area-scope link state advertisements

Returns:

- 0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.56 `int smi_ospf6_get_area_lsdb_advertisement (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, struct smi_ospf6_lsa_header * lsa, size_t * size)`

This function gets the entire link state advertisement, including its header. `smi_ospf6_get_area_lsdb_advertisement`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *arealsdbAdvertisement* Entire LSA including its header
- *size* Length of the LSA content

Returns:

- 0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.57 `int smi_ospf6_get_area_lsdb_age (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * areaLsdbAge)`

This function gets age field in link state advertisement which is the age of the link state advertisement in seconds. `smi_ospf6_get_area_lsdb_age`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *areaLsdbAge* Age field value in link state advertisement

Returns:

- 0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.58 `int smi_ospf6_get_area_lsdb_checksum (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * areaLsdbChecksum)`

This function gets checksum field which is the computed checksum of the complete contents of the link state advertisement, excepting the age field. The age field is excepted so that an advertisement's age can be incremented without updating the checksum. `smi_ospf6_get_area_lsdb_checksum`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *areaLsdbChecksum* Checksum vlaue in link state advertisement

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.59 `int smi_ospf6_get_area_lsdb_sequence (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * areaLsdbSequence)`

This function gets the sequence number field in Link state advertisement, is a signed 32-bit integer. It is used to detect old and duplicate link state advertisements. `smi_ospf6_get_area_lsdb_sequence`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *areaLsdbSequence* Sequence number field in link state advertisement

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.60 `int smi_ospf6_get_area_lsdb_type_known (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * areaLsdbTypeKnown)`

This function gets the value which indicates that the LSA(Link state advertisement) type is recognized by this router. `smi_ospf6_get_area_lsdb_type_known`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *areaLsdbTypeKnown* Flag to whether LSA type is recognized
 Recognized-OSPF6_API_TRUE
 Unrecognized-OSPF6_API_FALSE

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.61 `int smi_ospf6_get_area_nssa_trans_events (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int * areaNssaTransEvents)`

This function indicates the number of Translator state changes that have occurred since the last start-up of the OSPFv3 routing process. `smi_ospf6_get_area_nssa_trans_events`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- *areaNssaTransEvents* NSSA(Not-so-Stubby Area) transaltor events count

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.62 `int smi_ospf6_get_area_nssa_trans_role (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int * areaNssaTransRole)`

This function indicates an NSSA border router's policy to perform NSSA(Not-so-Stubby Area) translation of NSSA-LSAs into(Autonomous System) AS-External-LSAs. `smi_ospf6_get_area_nssa_trans_role`

Parameters:

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

- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- *areaNssaTransRole* NSSA Role during configuration
 - 1-OSPF_NSSA_TRANSLATE_ALWAYS
 - 2-OSPF_NSSA_TRANSLATE_CANDIDATE

Returns:

- 0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.63 int smi_ospf6_get_area_nssa_trans_stability_interval (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaNssaTransStabilityInterval)

This function gets the stability interval defined as the number of seconds after an elected translator determines its services are no longer required that it should continue to perform its translation duties. smi_ospf6_get_area_nssa_trans_stability_interval

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- *areaNssaTransStabilityInterval* Stability interval value

Returns:

- 0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.64 int smi_ospf6_get_area_nssa_trans_state (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *areaNssaTransState)

This function indicates if and how an NSSA border router is performing NSSA(Not-so-Stubby Area) translation of NSSA-LSAs into AS-External-LSAs. smi_ospf6_get_area_nssa_trans_state

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- *areaNssaTransState* Status of NSSA translator
 - 0-OSPF_NSSA_TRANSLATOR_DISABLED
 - 1-OSPF_NSSA_TRANSLATOR_ENABLED
 - 2-OSPF_NSSA_TRANSLATOR_ELECTED

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.65 `int smi_ospf6_get_area_status (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int * areaStatus)`

This function gets object which permits management of the table by facilitating actions such as row creation, construction, and destruction. `smi_ospf6_get_area_status`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 Process tag
- ← **areaId** OSPFv3 area ID in an IPv4 address format
- **areaStatus** Object which permits management of table

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.66 `int smi_ospf6_get_area_stub_metric_type (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int * areaStubMetricType)`

This function gets the type of metric advertised as a default route. `smi_ospf6_get_area_stub_metric_type`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 Process tag
- ← **areaId** OSPFv3 area ID in an IPv4 address format
- **areaStubMetricType** Metric type
2-if Metric type is EXTERNAL_METRIC_TYPE_1 3-if Metric type is EXTERNAL_METRIC_TYPE_2

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.67 `int smi_ospf6_get_area_summary (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int * areaSummary)`

This function gets the variable `ospfv3AreaSummary` controls the import of Inter-Area LSAs into stub and NSSA areas. `smi_ospf6_get_area_summary`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- *areaSummary* Value of ospfv3AreaSummary
 - 1-OSPF6_NO_AREA_SUMMARY
 - 2-OSPF6_SEND_AREA_SUMMARY

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.68 `int smi_ospf6_get_area_te_enabled (struct smiclient_globals *
azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int *
areaTeEnabled)`

This function indicates whether or not traffic engineering is enabled in the area. smi_ospf6_get_area_te_enabled

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- *areaTeEnabled* Traffic engineering status
 - Enabled-OSPF6_API_TRUE
 - Disabled-OSPF6_API_FALSE

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.69 `int smi_ospf6_get_as_lsdb_advertisement (struct smiclient_globals
* azg, char * ospf6ProcessTag, int type, struct pal_in4_addr
routerId, struct pal_in4_addr lsId, struct smi_ospf6_lsa_header *
asLsdbAdvertisement, size_t * size)`

This function gets the entire link state advertisement including the header. smi_ospf6_get_as_lsdb_advertisement

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *type* Link State advertisement type

- ← *routerId* Router ID in an IPv4 format
- ← *lsId* Link State ID
- *asLsdbAdvertisement* Link state advertisement data
- *size* Link state advertisement data length

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.70 `int smi_ospf6_get_as_lsdb_age (struct smiclient_globals * azg, char * ospf6ProcessTag, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * asLsdbAge)`

This function gets the age of the link state advertisement in seconds. `smi_ospf6_get_as_lsdb_age`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *type* Link State advertisement type
- ← *routerId* Router ID in an IPv4 format
- ← *lsId* Link State ID
- *asLsdbAge* link state advertisement age in seconds

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.71 `int smi_ospf6_get_as_lsdb_checksum (struct smiclient_globals * azg, char * ospf6ProcessTag, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * asAsdbChecksum)`

This function gets the checksum of the complete contents of the advertisement, excepting the age field. The age field is excepted so that an advertisement's age can be incremented without updating the checksum. `smi_ospf6_get_as_lsdb_checksum`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *type* Link State advertisement type
- ← *routerId* Router ID in an IPv4 format
- ← *lsId* Link State ID

→ *asLsdbChecksum* Value of Checksum applied

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.72 `int smi_ospf6_get_as_lsdb_sequence (struct smiclient_globals * azg, char * ospf6ProcessTag, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * asLsdbSequence)`

This function gets the sequence number field is a signed 32-bit integer. It is used to detect old and duplicate link state advertisements. smi_ospf6_get_as_lsdb_sequence

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *tag* OSPFv3 Process tag
 ← *routerId* Router ID in an IPv4 format
 ← *lsId* Link State ID
 → *asLsdbSequence* Sequence number field value

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.73 `int smi_ospf6_get_as_lsdb_type_known (struct smiclient_globals * azg, char * ospf6ProcessTag, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * asLsdbTypeKnown)`

This function gets value which indicates that the LSA(Link state advertisement) type is recognized by this router. smi_ospf6_get_as_lsdb_type_known

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *tag* OSPFv3 Process tag
 ← *type* Link State advertisement type
 ← *routerId* Router ID in an IPv4 format
 ← *lsId* Link State ID
 → *asLsdbTypeKnown* Flag to whether LSA type is recognized
 Recognized-OSPF6_API_TRUE Unrecognized-OSPF6_API_FALSE

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.74 **int smi_ospf6_get_as_scope_lsa_cksumsum** (struct smiclient_globals * *azg*, char * *ospf6ProcessTag*, int * *asScopeLsaCksumsum*)

This function gets the 32-bit unsigned sum of the LS(Link-state) checksums of the AS-scoped link state advertisements contained in the link state database. smi_ospf6_get_as_scope_lsa_cksumsum

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *asScopeLsaCksumsum* Sum of the LS(Link-state)checksums

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.75 **int smi_ospf6_get_as_scope_lsa_count** (struct smiclient_globals * *azg*, char * *ospf6ProcessTag*, int * *asScopeLsaCount*)

This function gets the number of (Autonomous System)AS-Scope (e.g. AS-External) link state advertisements in the link state database. smi_ospf6_get_as_scope_lsa_count

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *asScopeLsaCount* The number of AS-Scope LSA(Link state advertisements)

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.76 **int smi_ospf6_get_asbdr_rtr_count** (struct smiclient_globals * *azg*, char * *ospf6ProcessTag*, struct pal_in4_addr *areaId*, int * *asbdrRtrCount*)

This function gets the total number of Autonomous System border routers reachable within this area. This is initially zero, and is calculated in each SPF pass. smi_ospf6_get_asbdr_rtr_count

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

→ *asbdrRtrCount* Total number of Autonomous System border routers reachable

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.77 `int smi_ospf6_get_asbdr_rtr_status (struct smiclient_globals * azg,
char * ospf6ProcessTag, int * asBdrRtrStatus)`

This function gets flag to note whether this router is configured as an Autonomous System border router. smi_ospf6_get_asbdr_rtr_status

Parameters:

← *azg* Pointer to the SMI client global structure

← *tag* OSPFv3 Process tag

→ *asBdrRtrStatus* Flag value (OSPF6_API_TRUE|OSPF6_API_FALSE)

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.78 `int smi_ospf6_get_discontinuity_time (struct smiclient_globals * azg,
char * ospf6ProcessTag, int * discontinuityTime)`

This function gets the value of sysUpTime counter, which is the most recent time at which any one of this MIBs counters experienced a discontinuity. smi_ospf6_get_discontinuity_time

Parameters:

← *azg* Pointer to the SMI client global structure

← *tag* OSPFv3 Process tag

→ *discontinuityTime* Value of sysUpTime counter

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.79 `int smi_ospf6_get_extern_lsa_count (struct smiclient_globals * azg,
char * ospf6ProcessTag, int * externLsaCount)`

This function gets the number of External(LS(Link-state) type 0x4005) in the link state database. smi_ospf6_get_extern_lsa_count

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *externLsaCount* Number of External LS(Link-state)type

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.80 `int smi_ospf6_get_if_admin_stat (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifAdminStat)`

This function gets the administrative status of the given OSPFv3 interface. `smi_ospf6_get_if_admin_stat`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifAdminStat* Administrative status of the OSPFv3 interface
 Enabled-OSPF6_API_STATUS_ENABLED Disabled-OSPF6_API_STATUS_DISABLED

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.81 `int smi_ospf6_get_if_area_id (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr * ifAreaId)`

This function gets the area ID, which is a 32-bit integer uniquely identifying the area to which the given interface connects. Area ID 0 is used for the OSPFv3 backbone. `smi_ospf6_get_if_area_id`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifAreaId* Area identifier

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

2.1.2.82 `int smi_ospf6_get_if_bdr (struct smiclient_globals * azg, char *
ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr *
ifBdr)`

This function gets the interface address of the Backup Designated Router (BDR). smi_ospf6_get_if_bdr

Parameters:

← *azg* Pointer to the SMI client global structure
← *tag* OSPFv3 process tag
← *ifindex* Interface index
← *instanceId* Instance identifier
→ *ifBdr* Interface address of the BDR

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

2.1.2.83 `int smi_ospf6_get_if_demand (struct smiclient_globals * azg, char *
ospf6ProcessTag, int ifindex, int instanceId, int * ifDemand)`

This function gets the value that indicates whether Demand OSPFv3 procedures (Hello suppression to FULL neighbors and setting the DoNotAge flag on propagated LSAs) are performed on the given interface. smi_ospf6_get_if_demand

Parameters:

← *azg* Pointer to the SMI client global structure
← *tag* OSPFv3 process tag
← *ifindex* Interface index
← *instanceId* Instance identifier
→ *ifDemand* Value indicating whether Demand OSPFv3 procedures are performed on the given interface: OSPF6_API_FALSE -Disabled. This is the default value.

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

2.1.2.84 int smi_ospf6_get_if_demand_nbr_probe (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifDemandNbrProbe)

This function gets the value that indicates whether neighbor probing is enabled to determine whether the neighbor is inactive. If neighbor probing is disabled, the neighbor is inactive. smi_ospf6_get_if_demand_nbr_probe

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **ifindex** Interface index
- ← **instanceId** Instance identifier
- **ifDemandNbrProbe** Neighbor probe value: OSPF6_API_FALSE Disabled. This is the default value.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.85 int smi_ospf6_get_if_demand_nbr_probe_interval (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifDemandNbrProbeInterval)

This function gets the number of times the neighbor is to be probed. smi_ospf6_get_if_demand_nbr_probe_interval

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **ifindex** Interface index
- ← **instanceId** Instance identifier
- **ifDemandNbrProbeInterval** Neighbor probe interval. The default value is 120.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.86 int smi_ospf6_get_if_demand_nbr_probe_retrans_limit (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifDemandNbrProbeRetransLimit)

This function gets the configured number of consecutive LSA retransmissions permitted before the neighbor is considered inactive and the neighbor adjacency is brought down. smi_ospf6_get_if_demand_nbr_probe_retrans_limit

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifDemandNbrProbeRetransLimit* Number of consecutive LSA retransmissions permitted. The default value is 10.

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

2.1.2.87 `int smi_ospf6_get_if_dr (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr * ifDr)`

This function gets the router identifier of the Designated Router for the given OSPFv3 interface. smi_ospf6_get_if_dr

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifDr* Router identifier of the Designated Router

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

2.1.2.88 `int smi_ospf6_get_if_events (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifEvents)`

This function gets the number of times the given OSPFv3 interface has changed its state or an error has occurred. smi_ospf6_get_if_events

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index of this OSPFv3 interface
- ← *instanceId* Instance identifier

→ *ifEvents* Number of events

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

**2.1.2.89 int smi_ospf6_get_if_hello_interval (struct smiclient_globals *
azg, char * ospf6ProcessTag, int ifindex, int instanceId, int *
ifHelloInterval)**

This function gets the length of time, in seconds, between the Hello packets that the router sends on the given interface. This value must be the same for all routers attached to a common network. smi_ospf6_get_if_hello_interval

Parameters:

← *azg* Pointer to the SMI client global structure
← *tag* OSPFv3 process tag
← *ifindex* Interface index
← *instanceId* Instance identifier
→ *ifHelloInterval* Interface hello interval in seconds

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

**2.1.2.90 int smi_ospf6_get_if_link_lsa_cksumsum (struct smiclient_globals *
azg, char * ospf6ProcessTag, int ifindex, int instanceId, int *
ifLinkLsaCksumsum)**

This function gets the 32-bit unsigned sum of the Link-Scope LSAs' LS checksums contained in the given link's LSDB. smi_ospf6_get_if_link_lsa_cksumsum

Parameters:

← *azg* Pointer to the SMI client global structure
← *tag* OSPFv3 process tag
← *ifindex* Interface index
← *instanceId* Instance identifier
→ *ifLinkLsaCksumsum* The 32-bit unsigned sum of the Link-Scope LSAs' LS checksums

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

2.1.2.91 `int smi_ospf6_get_if_link_lsa_suppression (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifLinkLsaSuppression)`

This function gets the value that indicates whether link LSA origination is suppressed for broadcast or NBMA interface types. `smi_ospf6_get_if_link_lsa_suppression`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifLinkLsaSuppression* Suppression value for the Link LSA origination: OSPF6_API_FALSE Not Suppressed OSPF6_API_TRUE Suppressed.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.92 `int smi_ospf6_get_if_link_scope_lsa_count (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifLinkScopeLsaCount)`

This function gets the total number of Link-Scope LSAs in the given link's LSDB. `smi_ospf6_get_if_link_scope_lsa_count`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifLinkScopeLsaCount* Total number of Link-Scope LSAs

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.93 `int smi_ospf6_get_if_metric_value (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifMetricValue)`

This function gets the metric value assigned to the given interface. `smi_ospf6_get_if_metric_value`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifMetricValue* Metric value

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.94 **int smi_ospf6_get_if_poll_interval (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifPollInterval)**

This function gets the larger time interval, in seconds, between the Hello packets sent to an inactive, non-broadcast multi-access neighbor for the given interface. smi_ospf6_get_if_poll_interval

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifPollInterval* Poll interval for the interface: OSPF_POLL_INTERVAL_DEFAULT-Default value

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.95 **int smi_ospf6_get_if_retrans_interval (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifRetransInterval)**

This function gets the number of seconds between LSA retransmissions for adjacencies belonging to the given interface. smi_ospf6_get_if_retrans_interval

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index

- ← *instanceId* Instance identifier
- *ifRtrInterval* Interval between LSA retransmissions

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.96 int smi_ospf6_get_if_rtr_dead_interval (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifRtrDeadInterval)

This function gets the number of seconds that a router's Hello packets have not been seen before its neighbors declare the router down on the given interface. smi_ospf6_get_if_rtr_dead_interval

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifRtrDeadInterval* Router dead interval, in seconds

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.97 int smi_ospf6_get_if_rtr_priority (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifRtrPriority)

This function gets the priority value of the given interface. A value of 0 signifies that the router is not eligible to become the Designated Router on this particular network. In the event of a tie in this value, the routers use their Router identifier as a tie breaker. smi_ospf6_get_if_rtr_priority

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifRtrPriority* Interface priority value

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.98 `int smi_ospf6_get_if_state (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifState)`

This function gets the state of the given OSPFv3 interface. `smi_ospf6_get_if_state`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **ifindex** Interface index
- ← **instanceId** Instance identifier
- **ifState** State of the OSPFv3 interface: Interface is down, Interface is a loopback interface, Interface is in a waiting state, Interface is a point-to-point interface, Interface is a Designated Router, Interface is a Backup Designated Router, Interface is an Other Designated Router, Interface is in a standby state.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.99 `int smi_ospf6_get_if_status (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifStatus)`

This function gets the status of whether OSPFv3 is configured on the given interface. `smi_ospf6_get_if_status`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **ifindex** Interface index
- ← **instanceId** Instance identifier
- **ifStatus** Row status of the interface: ROW_STATUS_ACTIVE-Row is active, ROW_STATUS_NONEXISTENT Row is nonexistent.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.100 `int smi_ospf6_get_if_te_disabled (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifTeDisabled)`

This function gets the value that indicates whether traffic engineering is disabled on the given interface when traffic engineering is enabled in the area to which the interface is attached. `smi_ospf6_get_if_te_disabled`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifTeDisabled* Status of traffic engineering:OSPF6_API_FALSE Enabled. This is the default value. OSPF6_API_TRUE Disabled.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.101 `int smi_ospf6_get_if_transit_delay (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifTransitDelay)`

This function gets the transit-delay value of the given OSPFv3 interface. This value is an estimate of the number of seconds required to transmit a link-state update packet through the interface. smi_ospf6_get_if_transit_delay

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- *ifTransitDelay* Transit-delay value, in seconds

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.102 `int smi_ospf6_get_if_type (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int * ifType)`

This function gets the interface type of the given OSPFv3 interface. smi_ospf6_get_if_type

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index

- ← *instanceId* Instance identifier
- *ifType* Interface types: Broadcast, Non-broadcast multi-access (NBMA), Point-to-Point, Point-to-Multipoint

Returns:

0 on success, otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.103 `int smi_ospf6_get_import_as_extern (struct smiclient_globals *azg, char *ospf6ProcessTag, struct pal_in4_addr areaId, int *importAsExtern)`

This function indicates whether an area is a stub area, NSSA, or standard area. `smi_ospf6_get_import_as_extern`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- *importAsExtern* Area status
 - Stub area-OSPF6_API_IMPORT_NO_EXTERNAL
 - NSSA area-OSPF6_API_IMPORT_NSSA
 - Standard area-OSPF6_API_IMPORT_EXTERNAL

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.104 `int smi_ospf6_get_link_lsdb_advertisement (struct smiclient_globals *azg, char *ospf6ProcessTag, int ifindex, int instanceId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, struct smi_ospf6_lsa_header *lsa, size_t *size)`

This function gets the entire LSA, including its header, from the local Link-Scope LSDB for the given interface. `smi_ospf6_get_link_lsdb_advertisement`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *ifindex* Identifier of the link from which the LSA was received
- ← *instanceId* OSPFv3 Instance identifier
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB

- ← ***routerId*** Router identifier of the originating router
- ← ***lsId*** Link state identifier of the LSA
- ***linkLsdbAdvertisement*** Entire LSA including its header
- ***size*** Length of the LSA content

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.105 `int smi_ospf6_get_link_lsdb_age (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * linkLsdbAge)`

This function gets the age of the LSA that is stored in the local Link-Scope LSDB.
smi_ospf6_get_link_lsdb_age

Parameters:

- ← ***azg*** Pointer to the SMI client global structure
- ← ***tag*** OSPFv3 Process tag
- ← ***ifindex*** Identifier of the link from which the LSA was received
- ← ***instanceId*** OSPFv3 Instance identifier
- ← ***type*** Type of LSA that is stored in the local Link-Scope LSDB
- ← ***routerId*** Router identifier of the originating router
- ← ***lsId*** Link state identifier of the LSA
- ***linkLsdbAge*** Age of the LSA in seconds

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.106 `int smi_ospf6_get_link_lsdb_checksum (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * linkLsdbChecksum)`

This function gets the checksum of the contents of the LSA that is stored in the specified local Link-Scope LSDB, excluding the age field, for the given OSPFv3 interface. smi_ospf6_get_link_lsdb_checksum

Parameters:

- ← ***azg*** Pointer to the SMI client global structure
- ← ***tag*** OSPFv3 Process tag
- ← ***ifindex*** Identifier of the link from which the LSA was received

- ← *instanceId* OSPFv3 Instance identifier
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *linkLsdbChecksum* Checksum of the contents of the LSA excluding the age field

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.107 `int smi_ospf6_get_link_lsdb_sequence (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * linkLsdbSequence)`

This function gets the LSDB sequence number of the local Link-Scope LSA for the given interface. The sequence number is used to detect old and duplicate LSAs. `smi_ospf6_get_link_lsdb_sequence`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *ifindex* Identifier of the link from which the LSA was received
- ← *instanceId* OSPFv3 Instance identifier
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *linkLsdbSequence* LSDB sequence number of the LSA

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.108 `int smi_ospf6_get_link_lsdb_type_known (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int type, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * linkLsdbTypeKnown)`

This function gets the value that indicates whether the LSA type is recognized by the given router. `smi_ospf6_get_link_lsdb_type_known`

Parameters:

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

- ← *tag* OSPFv3 Process tag
- ← *ifindex* Identifier of the link from which the LSA was received
- ← *instanceId* OSPFv3 Instance identifier
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *linkLsdbTypeKnown* Flag to whether LSA type is recognized
 - Recognized-OSPF6_API_TRUE
 - Unrecognized-OSPF6_API_FALSE

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.109 `int smi_ospf6_get_nbr_address (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, char * nbrAddress, size_t * size)`

This call gets the IPv6 address the neighbor associated with the local link. smi_ospf6_get_nbr_address

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* The local link ID of the link over which the neighbor can be reached
- ← *instanceId* Instance identifier
- ← *routerId* The router ID of the virtual neighbor
- *nbrAddress* The IPv6 address of ospfv3 neighbor
- *size* The length of IPv6 address of OSPFv3 neighbor

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

2.1.2.110 `int smi_ospf6_get_nbr_address_type (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int * nbrAddressType)`

This call gets the address type of ospfv3NbrAddress. smi_ospf6_get_nbr_address_type

Parameters:

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

- ← **tag** OSPFv3 process tag
- ← **ifindex** the local link ID of the link over which the neighbor can be reached
- ← **instanceId** Instance identifier
- ← **routerId** The router ID of the virtual neighbor
- **nbrAddressType** The address type of ospfv3NbrAddress

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.111 `int smi_ospf6_get_nbr_events (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int * nbrEvents)`

This call gets the number of times this neighbor relationship has changed state, or an error has occurred. `smi_ospf6_get_nbr_events`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **ifindex** The local link ID of the link over which the neighbor can be reached
- ← **instanceId** Instance identifier
- ← **routerId** The router ID of the virtual neighbor
- **nbrEvents** the number of times this neighbor relationship has changed state, or an error has occurred

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.112 `int smi_ospf6_get_nbr_hello_suppressed (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int * nbrHelloSuppressed)`

This call gets the indication whether Hellos are being suppressed to the neighbor. `smi_ospf6_get_nbr_hello_suppressed`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **ifindex** The local link ID of the link over which the neighbor can be reached

- ← *instanceId* Instance identifier
- ← *routerId* The router ID of the virtual neighbor
- *nbrHelloSuppressed* The indication whether Hellos are being suppressed to the neighbor. Hello suppression is not supported in the current implementation. Hence, returns False.

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.113 `int smi_ospf6_get_nbr_if_id (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int * nbrIfId)`

This call gets the interface ID that the neighbor advertises in its Hello Packets on this link. `smi_ospf6_get_nbr_if_id`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* The local link ID of the link over which the neighbor can be reached
- ← *instanceId* Instance identifier
- ← *routerId* The router ID of the virtual neighbor
- *nbrIfId* The interface ID that the neighbor advertises in its Hello Packets on this link.

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.114 `int smi_ospf6_get_nbr_lsretransq_len (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int * nbrLsretransqLen)`

This call gets the current length of the retransmission queue. `smi_ospf6_get_nbr_lsretransq_len`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* the local link ID of the link over which the neighbor can be reached
- ← *instanceId* Instance identifier

- ← **routerId** the router ID of the virtual neighbor.
- **nbrLsretransqLen** the current length of the retransmission queue.

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.115 `int smi_ospf6_get_nbr_options (struct smiclient_globals * azg, char *
 ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr
 routerId, int * nbrOptions)`

This call gets a Bit Mask corresponding to the neighbor's options field. smi_ospf6_get_nbr_options

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **ifindex** The local link ID of the link over which the neighbor can be reached
- ← **instanceId** Instance identifier
- ← **routerId** The router ID of the virtual neighbor
- **nbrOptions** a Bit Mask corresponding to the neighbor's options field

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.116 `int smi_ospf6_get_nbr_priority (struct smiclient_globals * azg, char *
 ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr
 routerId, int * nbrPriority)`

This call gets the priority of this neighbor in the DR election algorithm. smi_ospf6_get_nbr_priority

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **ifindex** The local link ID of the link over which the neighbor can be reached
- ← **instanceId** Instance identifier
- ← **routerId** The router ID of the virtual neighbor
- **nbrPriority** The priority of this neighbor in the DR election algorithm

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.117 `int smi_ospf6_get_nbr_restart_helper_age (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int * nbrRestartHelperAge)`

This function gets the remaining time in the current OSPF graceful restart interval for the given router that is acting as a restart helper for the neighbor.. smi_ospf6_get_nbr_restart_helper_age

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* the local link ID of the link over which the neighbor can be reached
- ← *instanceId* Instance identifier
- ← *routerId* Router identifier of the neighbor
- *nbrRestartHelperAge* Remaining time, in seconds, in the current OSPF graceful restart interval

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.118 `int smi_ospf6_get_nbr_restart_helper_exit_reason (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int * nbrRestartHelperExitReason)`

This function gets the outcome of the last attempt the given router was acting as a graceful restart helper for the neighbor. smi_ospf6_get_nbr_restart_helper_exit_reason

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* the local link ID of the link over which the neighbor can be reached
- ← *instanceId* Instance identifier
- ← *routerId* Router identifier of the neighbor
- *nbrRestartHelperExitReason* Exit reason:none- No restart has yet been attempted , inProgress- A restart attempt is currently underway,completed -No restart has yet been attempted,timedOut -Last time restart timed out,topology Changed -Last time restart was aborted due to a topology change.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.119 `int smi_ospf6_get_nbr_restart_helper_status (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int * nbrRestartHelperStatus)`

This function gets the value that indicates whether the given router is a graceful restart helper for the neighbor. `smi_ospf6_get_nbr_restart_helper_status`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **ifindex** The local link ID of the link over which the neighbor can be reached
- ← **instanceId** Instance identifier
- ← **routerId** Router identifier of the neighbor
- **nbrRestartHelperStatus** Neighbor restart helper status: OSPF6_API_RESTART_HELPER Graceful restart helper , SPF6_API_RESTART_HELPER_NONE Not a graceful restart helper.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.120 `int smi_ospf6_get_nbr_state (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, struct pal_in4_addr routerId, int * nbrState)`

This call gets the state of the relationship with this neighbor. `smi_ospf6_get_nbr_state`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **ifindex** the local link ID of the link over which the neighbor can be reached
- ← **instanceId** Instance identifier
- ← **routerId** The router ID of the virtual neighbor
- **nbrState** The state of the relationship with this Neighbor
down,attempt,init,twoWay,exchangeStart,exchange,loading,full.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.121 int smi_ospf6_get_notification_enable (struct smiclient_globals * *azg*, char * *ospf6ProcessTag*, int * *notificationEnable*)

This function gets the value that indicates whether the generation of OSPFv3 notifications is enabled. smi_ospf6_get_notification_enable

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *notificationEnable* Status of whether the generation of OSPFv3 notifications is enabled

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.122 int smi_ospf6_get_originate_new_lsas (struct smiclient_globals * *azg*, char * *ospf6ProcessTag*, int * *originateNewLsas*)

This function gets the number of new link-state advertisements that have been originated. This number is incremented each time the router originates a new LSA. smi_ospf6_get_originate_new_lsas

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *originateNewLsas* Number of new link-state advertisements

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.123 int smi_ospf6_get_reference_bandwidth (struct smiclient_globals * *azg*, char * *ospf6ProcessTag*, int * *referenceBandwidth*)

This function gets reference bandwidth in kilobits per second for calculating default interface metrics. smi_ospf6_get_reference_bandwidth

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *referenceBandwidth* Reference bandwidth in kilobits

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.124 `int smi_ospf6_get_restart_age (struct smiclient_globals * azg, char * ospf6ProcessTag, int * restartAge)`

This function gets remaining time, in seconds, in the current OSPF graceful restart interval for the router. `smi_ospf6_get_restart_age`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *restartAge* Remaining time, in seconds, in the current OSPF graceful restart interval

Returns:

- 0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.125 `int smi_ospf6_get_restart_exit_reason (struct smiclient_globals * azg, char * ospf6ProcessTag, int * restartExitReason)`

This function gets the outcome of the last attempt at a graceful restart for the router. `smi_ospf6_get_restart_exit_reason`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *restartExitReason* Exit reason
 - none-No restart has yet been attempted
 - inProgress-A restart attempt is currently underway
 - completed-Last restart completed successfully
 - timedOut-Last time restart timed out
 - topologyChanged-Last time restart was aborted due to a topology change

Returns:

- 0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.126 `int smi_ospf6_get_restart_interval (struct smiclient_globals * azg, char * ospf6ProcessTag, int * restartInterval)`

This function gets the interval of the graceful restart timeout for the router. `smi_ospf6_get_restart_interval`

Parameters:

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

- ← *tag* OSPFv3 Process tag
- *restartInterval* Interval of the graceful restart timeout

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.127 int smi_ospf6_get_restart_status (struct smiclient_globals * *azg*, char * *ospf6ProcessTag*, int * *restartStatus*)

This function gets the current status of OSPF graceful restart capability for the router. smi_ospf6_get_restart_status

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *restartStatus* Status of OSPF graceful restart
Unplanned-OSPF6_API_RESTART_UNPLANNED |
lanned-OSPF6_API_RESTART_PLANNED
None-OSPF6_API_RESTART_NONE

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.128 int smi_ospf6_get_restart_strict_lsa_check (struct smiclient_globals * *azg*, char * *ospf6ProcessTag*, int * *restartStrictLsaCheckStatus*)

This function gets the value that indicates whether strict LSA checking is enabled for graceful restart on the router. smi_ospf6_get_restart_strict_lsa_check

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *restartStrictLsaCheckStatus* Status of whether strict LSA checking is enabled
OSPF6_API_RESTART_LSA_CHECK_ENABLE
OSPF6_API_RESTART_LSA_CHECK_DISABLE

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.129 `int smi_ospf6_get_restart_support (struct smiclient_globals * azg, char * ospf6ProcessTag, int * restartSupport)`

This function gets the value that indicates whether the router supports OSPF graceful restart. `smi_ospf6_get_restart_support`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *restartSupport* Status of support for OSPF graceful restart
1-None, 3-Planned and Unplanned restart

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.130 `int smi_ospf6_get_restart_time (struct smiclient_globals * azg, char * ospf6ProcessTag, int * restartTime)`

This function gets the value of sysUpTime on the most recent time at which the `ospfv3RestartExitReason` object was updated. `smi_ospf6_get_restart_time`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *restartTime* Value of sysUpTime

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.131 `int smi_ospf6_get_router_id (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr * routerId)`

This function gets 32-bit IPv4 interface address uniquely identifying the router in the Autonomous System. `smi_ospf6_get_routerId`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *routerId* IPv4 address of router's OSPFv3 interface

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.132 `int smi_ospf6_get_rx_new_lsas (struct smiclient_globals * azg, char * ospf6ProcessTag, int * rxNewLsas)`

This function gets the number of link state advertisements received determined to be new instantiations. This number does not include newer instantiations of self-originated link state advertisements. `smi_ospf6_get_rx_new_lsas`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- *rxNewLsas* Number of new link-state advertisements received

Returns:

- 0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.133 `int smi_ospf6_get_spf_runs (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int * spf_runs)`

This function gets number of times that the intra-area route table has been calculated using this area's link state database. `smi_ospf6_get_spf_runs`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- *spf_runs* Number of times intra-route table has been calculated.

Returns:

- 0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.134 `int smi_ospf6_get_stub_metric (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int * stubMetric)`

This function gets the metric value advertised for the default route into stub and NSSA areas. By default, this equals the least metric among the interfaces to other areas. `smi_ospf6_get_stub_metric`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format

→ *stubMetric* The metric value

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.135 `int smi_ospf6_get_stub_router_advertisement (struct smiclient_globals * azg, char * ospf6ProcessTag, int * stubRouterAdvertisement)`

This function gets the value that indicates whether the router advertises stub or standard LSAs(Link state advertisements). `smi_ospf6_get_stub_router_advertisement`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *tag* OSPFv3 Process tag
 → *stubRouterAdvertisement* Type of stub router advertisements

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.136 `int smi_ospf6_get_stub_router_support (struct smiclient_globals * azg, char * ospf6ProcessTag, int * stubRouterSupport)`

This function gets the value that indicates whether the router supports the stub router functionality. `smi_ospf6_get_stub_router_support`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *tag* OSPFv3 Process tag
 → *stubRouterSupport* Status for support of stub router functionality:

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.137 `int smi_ospf6_get_version_num (struct smiclient_globals * azg, char * ospf6ProcessTag, int * versionNum)`

This function gets the version number of OSPF for IPv6. `smi_ospf6_get_version_num`

Parameters:

← *azg* Pointer to the SMI client global structure

← *tag* OSPFv3 Process tag

→ *versionNum* Version no (for OSPF IPV6 version no is 3)

Returns:

0 on success, otherwise OSPF6_API_GET_ERROR

2.1.2.138 `int smi_ospf6_get_virt_if_events (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfEvents)`

This call gets the number of state changes or error events on this Virtual Link. `smi_ospf6_get_virt_if_events`

Parameters:

← *azg* Pointer to the SMI client global structure

← *tag* OSPFv3 process tag

← *areaId* The Transit Area ID.

← *routerId* the router ID of the virtual neighbor.

→ *virtIfEvents* the number of state changes or error events on this Virtual Link.

Returns:

0 on success , otherwise the following error codes

OSPF6_API_GET_ERROR

2.1.2.139 `int smi_ospf6_get_virt_if_hello_interval (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfHelloInterval)`

This call gets the length of time, in seconds, between the Hello packets that the router sends on the virtual interface. `smi_ospf6_get_virt_if_hello_interval`

Parameters:

← *azg* Pointer to the SMI client global structure

← *tag* OSPFv3 process tag

← *areaId* The Transit Area ID

← *routerId* The router ID of the virtual neighbor

→ *virtIfHelloInterval* The number of seconds between the Hello packets that the router sends on the interface. The default value is 10 seconds

Returns:

0 on success , otherwise the following error codes

OSPF6_API_GET_ERROR

2.1.2.140 `int smi_ospf6_get_virt_if_index (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfIndex)`

This call gets the interface ID assigned to this OSPFv3 virtual interface. `smi_ospf6_get_virt_if_index`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID
- ← *routerId* The router ID of the virtual neighbor.
- *virtIfIndex* The interface ID assigned to this OSPFv3 virtual interface

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

2.1.2.141 `int smi_ospf6_get_virt_if_instid (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfInstanceId)`

This function gets the local Interface Instance identifier assigned by the OSPFv3 process to the given OSPFv3 virtual interface. `smi_ospf6_get_virt_if_instid`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID
- ← *routerId* Router identifier of the originating router
- *virtIfInstanceId* Local Interface Instance identifier of the OSPFv3 virtual interface

Returns:

0 on success , otherwise the following error codes
OSPF6_API_GET_ERROR

2.1.2.142 `int smi_ospf6_get_virt_if_link_lsa_cksumsum (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfLinkLsaChecksumsum)`

This call gets the 32-bit unsigned sum of the Link-Scope link-state advertisements' LS checksums contained in this virtual link's link-state database. `smi_ospf6_get_virt_if_link_lsa_cksumsum`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID.
- ← *routerId* The router ID of the virtual neighbor.
- *virtIfLinkLsaChecksum* The 32-bit unsigned sum of the Link-Scope link-state advertisements' LS checksums contained in this virtual link's link-state database

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.143 `int smi_ospf6_get_virt_if_link_scope_lsa_count (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfLinkScopeLsaCount)`

This call gets the total number of Link-Scope link-state advertisements in this virtual link's link-state database. `smi_ospf6_get_virt_if_link_scope_lsa_count`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID.
- ← *routerId* The router ID of the virtual neighbor.
- *virtIfLinkScopeLsaCount* The total number of Link-Scope link-state advertisements in this virtual link's link-state database.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.144 `int smi_ospf6_get_virt_if_retrans_interval (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfRetransInterval)`

This call gets the number of seconds between link-state advertisement retransmissions, for adjacencies belonging to this virtual interface. `smi_ospf6_get_virt_if_retrans_interval`

Parameters:

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

- ← **tag** OSPFv3 process tag
- ← **areaId** The Transit Area ID
- ← **routerId** The router ID of the virtual neighbor
- **virtIfRetransInterval** The number of seconds between link-state advertisement retransmissions, for adjacencies belonging to this interface. The default value is 5 seconds

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.145 `int smi_ospf6_get_virt_if_rtr_dead_interval (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfRtrDeadInterval)`

This call gets the number of seconds that a router's Hello packets have not been seen on this virtual interface, before its neighbors declare the router down. `smi_ospf6_get_virt_if_rtr_dead_interval`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **areaId** The Transit Area ID
- ← **routerId** The router ID of the virtual neighbor
- **virtIfRtrDeadInterval** The number of seconds that a router's Hello packets have not been seen before its neighbors declare the router down. The default value is 60 seconds

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.146 `int smi_ospf6_get_virt_if_state (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfState)`

This call gets the OSPF virtual interface state. `smi_ospf6_get_virt_if_state`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **areaId** The Transit Area ID

- ← **routerId** The router ID of the virtual neighbor
- **virtIfState** The OSPF virtual interface states down -- these use the same encoding, pointToPoint -- as the ospfv3IfTable, Loopback, Waiting, Designated Router, Backup Designated Router, Other Designated Router.

Returns:

0 on success, otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.147 `int smi_ospf6_get_virt_if_status (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfStatus)`

This call gets the status of the specified virtual interface. smi_ospf6_get_virt_if_status

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **areaId** The Transit Area ID.
- ← **routerId** The router ID of the virtual neighbor.
- **virtIfStatus** The status of the specified virtual interface

Returns:

0 on success, otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.148 `int smi_ospf6_get_virt_if_transmit_delay (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtIfTransmitDelay)`

This call gets the estimated number of seconds it takes to transmit a link-state update packet over this virtual interface. smi_ospf6_get_virt_if_transmit_delay

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **areaId** The Transit Area ID
- ← **routerId** The router ID of the virtual neighbor
- **virtIfTransmitDelay** The estimated number of seconds it takes to transmit a link-state update packet over this interface. The default value is 1 second

Returns:

0 on success, otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.149 `int smi_ospf6_get_virt_link_lsdb_advertisement (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int ospf6LsaType, struct pal_in4_addr routerId, struct pal_in4_addr lsId, struct smi_ospf6_lsa_header * virtLinkLsdbAdvertisement, size_t * size)`

This function gets the entire LSA, including its header, from the local Link-Scope LSDB for the OSPFv3 virtual link. `smi_ospf6_get_virt_link_lsdb_advertisement`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* Area identifier in an IPv4 address format
- ← *nbrId* Neighbor process identifier
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *virtLinkLsdbAdvertisement* Entire LSA including its header
- *size* Length of the LSA content

Returns:

- 0 on success, otherwise following error codes `OSPF6_API_GET_ERROR`

2.1.2.150 `int smi_ospf6_get_virt_link_lsdb_age (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int ospf6LsaType, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * virtLinkLsdbAge)`

This function gets the age of the LSA that is stored in the local Link-Scope LSDB for the given virtual OSPFv3 interface. `smi_ospf6_get_virt_link_lsdb_age`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* Area identifier in an IPv4 address format
- ← *nbrId* Neighbor process identifier
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *virtLinkLsdbAge* Age of the LSA, in seconds

Returns:

- 0 on success, otherwise following error codes `OSPF6_API_GET_ERROR`

2.1.2.151 `int smi_ospf6_get_virt_link_lsdb_checksum (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int ospf6LsaType, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * virtLinkLsdbChecksum)`

This function gets the entire LSA, including its header, from the local Link-Scope LSDB for the OSPFv3 virtual link. `smi_ospf6_get_virt_link_lsdb_checksum`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* Area identifier in an IPv4 address format
- ← *nbrId* Neighbor process identifier
- ← *type* Type of LSA that is stored in the local Link-Scope LSDB
- ← *routerId* Router identifier of the originating router
- ← *lsId* Link state identifier of the LSA
- *virtLinkLsdbChecksum* Checksum of the complete contents of the LSA excluding the age field

Returns:

- 0 on success, otherwise following error codes `OSPF6_API_GET_ERROR`

2.1.2.152 `int smi_ospf6_get_virt_link_lsdb_sequence (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int ospf6LsaType, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * virtLinkLsdbSequence)`

This function gets the aggregate route tag, which is advertised only in the summarized As-External LSA when summarizing from NSSA-LSAs to AS-External-LSAs. `smi_ospf6_get_area_aggregate_route_tag`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* Area identifier in an IPv4 address format
- ← *areaLsdbType* Area LSDB type that the given Address Aggregate applies to
- ← *prefixType* Prefix type of `ospfv3AreaAggregatePrefix` object
- ← *prefix* IPv6 prefix
- ← *prefixlen* Length of the prefix (in bits). A prefix can not be shorter than 3 bits.
- *areaAggregateRouteTag* Aggregate route tag

Returns:

- 0 on success, otherwise following error codes `OSPF6_API_GET_ERROR`

`smi_ospf6_get_virt_link_lsdb_sequence` This function gets the LSDB sequence number of the local Link-Scope LSA for the given virtual interface. The sequence number is used to detect old and duplicate LSAs.

Parameters:

- ← ***azg*** Pointer to the SMI client global structure
- ← ***tag*** OSPFv3 process tag
- ← ***areaId*** Area identifier in an IPv4 address format
- ← ***nbrId*** Neighbor process identifier
- ← ***type*** Type of LSA that is stored in the local Link-Scope LSDB
- ← ***routerId*** Router identifier of the originating router
- ← ***lsId*** Link state identifier of the LSA
- ***virtLinkLsdbSequence*** LSDB sequence number of the LSA

Returns:

- 0 on success, otherwise following error codes OSPF6_API_GET_ERROR

2.1.2.153 `int smi_ospf6_get_virt_link_lsdb_type_known (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int ospf6LsaType, struct pal_in4_addr routerId, struct pal_in4_addr lsId, int * virtLinkLsdbTypeKnown)`

This function gets the value that indicates whether the LSA type is recognized by the given router. `smi_ospf6_get_virt_link_lsdb_type_known`

Parameters:

- ← ***azg*** Pointer to the SMI client global structure
- ← ***tag*** OSPFv3 process tag
- ← ***areaId*** Area identifier in an IPv4 address format
- ← ***nbrId*** Neighbor process identifier
- ← ***type*** Type of LSA that is stored in the local Link-Scope LSDB
- ← ***routerId*** Router identifier of the originating router
- ← ***lsId*** Link state identifier of the LSA
- ***virtLinkLsdbTypeKnown*** Status of whether the LSA type is recognized:
 OSPF6_API_FALSE : Not recognized
 OSPF6_API_TRUE : Recognized

Returns:

- 0 on success, otherwise following error codes OSPF6_API_GET_ERROR

2.1.2.154 `int smi_ospf6_get_virt_nbr_address (struct smiclient_globals *
azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct
pal_in4_addr routerId, char * virtNbrAddress, size_t * size)`

This call gets the IPv6 address of the virtual neighbor associated with the local link. It must be a Site-Local or Global scope address. `smi_ospf6_get_virt_nbr_address`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID.
- ← *routerId* The router ID of the virtual neighbor
- *virtNbrAddress* The IPv6 address of virtual neighbor.
- *size* The length of the IPv6 address of virtual neighbor

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.155 `int smi_ospf6_get_virt_nbr_address_type (struct smiclient_globals *
azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct
pal_in4_addr routerId, int * virtNbrAddressType)`

This call gets the address type of `ospfv3VirtNbrAddress`. `smi_ospf6_get_virt_nbr_address_type`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID.
- ← *routerId* The router ID of the virtual neighbor
- *virtNbrAddressType* The address type of `ospfv3VirtNbrAddress`.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.156 `int smi_ospf6_get_virt_nbr_events (struct smiclient_globals *
azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct
pal_in4_addr routerId, int * virtNbrEvents)`

This call gets the number of times this virtual neighbor has changed its state, or an error has occurred. `smi_ospf6_get_virt_nbr_events`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* the Transit Area ID.
- ← *routerId* The router ID of the virtual neighbor
- *virtNbrEvents* The number of times this virtual neighbor has changed its state, or an error has occurred

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.157 `int smi_ospf6_get_virt_nbr_hello_suppressed (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtNbrHelloSuppressed)`

This call gets the indication whether Hellos are being suppressed to the virtual neighbor. `smi_ospf6_get_virt_nbr_hello_suppressed`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID
- ← *routerId* The router ID of the virtual neighbor
- *virtNbrHelloSuppressed* The indication whether Hellos are being suppressed to the virtual neighbor. Hello suppression is not supported in the current implementation. Hence, returns False.

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.158 `int smi_ospf6_get_virt_nbr_if_id (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtNbrIfId)`

This call gets the interface ID that the neighbor advertises in its Hello Packets on this virtual link. `smi_ospf6_get_virt_nbr_if_id`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag

- ← *areaId* The Transit Area ID
- ← *routerId* The router ID of the virtual neighbor
- *virtNbrIfId* The interface ID that the neighbor advertises in its Hello Packets on this virtual link

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.159 `int smi_ospf6_get_virt_nbr_ifindex (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtNbrIfindex)`

This function gets the priority number of the neighbor in the Designated Router Election algorithm. This number is used in the Designated Router Election algorithm to determine whether the neighbor is eligible to become the Designated Router on a given network. The value 0 disqualifies the router. `smi_ospf6_get_cfg_nbr_priority`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- ← *nbrAddrType* Neighbor address type
- ← *nbrAddr* IP address of the neighbor
- *cfgNbrPriority* Priority number of the neighbor

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

`smi_ospf6_get_virt_nbr_ifindex` This call gets the local interface ID for the virtual link over which the neighbor can be reached.

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID
- ← *routerId* The router ID of the virtual neighbor
- *virtNbrIfindex* The local interface ID for the virtual link over which the neighbor can be reached.

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.160 `int smi_ospf6_get_virt_nbr_ifinstid (struct smiclient_globals *
azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct
pal_in4_addr routerId, int * virtNbrIfInstanceId)`

This function gets the interface instance identifier of the given virtual interface over which the neighbor can be reached. smi_ospf6_get_virt_nbr_if_instid

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID.
- ← *routerId* The router ID of the virtual neighbor
- *virtNbrIfInstanceId* Interface instance identifier

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.161 `int smi_ospf6_get_virt_nbr_ls_retransq_len (struct smiclient_globals
* azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct
pal_in4_addr routerId, int * virtNbrLsRetransqLen)`

This call gets the current length of the retransmission queue. smi_ospf6_get_virt_nbr_ls_retransq_len

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* the Transit Area ID
- ← *routerId* The router ID of the virtual neighbor
- *virtNbrLsRetransqLen* The current length of the retransmission queue

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.162 `int smi_ospf6_get_virt_nbr_options (struct smiclient_globals *
azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct
pal_in4_addr routerId, int * virtNbrOptions)`

This call gets a Bit Mask corresponding to the virtual neighbor's options field. smi_ospf6_get_virt_nbr_options

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID
- ← *routerId* The router ID of the virtual neighbor
- *virtNbrOptions* A Bit Mask corresponding to the virtual neighbor's options field

Returns:

- 0 on success , otherwise the following error codes
- OSPF6_API_GET_ERROR

2.1.2.163 `int smi_ospf6_get_virt_nbr_restart_helper_age (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtNbrRestartHelperAge)`

This function gets the remaining time in the current OSPF graceful restart interval for the given virtual router that is acting as a restart helper for the neighbor. `smi_ospf6_get_virt_nbr_restart_helper_age`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *areaId* Area identifier in an IPv4 address format
- ← *routerId* Router identifier of the originating router
- *virtNbrRestartHelperAge* Remaining time, in seconds, in the current OSPF graceful restart interval

Returns:

- 0 on success, otherwise following error codes OSPF6_API_GET_ERROR

2.1.2.164 `int smi_ospf6_get_virt_nbr_restart_helper_exit_reason (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtNbrRestartHelperExitReason)`

This function gets the outcome of the last attempt of the given virtual router at acting as a graceful restart helper for the neighbor. `smi_ospf6_get_virt_nbr_restart_helper_exit_reason`

Parameters:

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

- ← **tag** OSPFv3 process tag
- ← **areaId** Area identifier in an IPv4 address format
- ← **routerId** Router identifier of the originating router
- **virtNbrRestartHelperExitReason** Exit reason: none : No restart has yet been attempted
inProgress : A restart attempt is currently underway completed : No restart has yet been attempted
timedOut : Last time restart timed out
topologyChanged : Last time restart was aborted due to a topology change

Returns:

0 on success, otherwise following error codes OSPF6_API_GET_ERROR

2.1.2.165 `int smi_ospf6_get_virt_nbr_restart_helper_status (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtNbrRestartHelperStatus)`

This function gets the value that indicates whether the virtual router is acting as a graceful restart helper for the neighbor. `smi_ospf6_get_virt_nbr_restart_helper_status`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 process tag
- ← **areaId** Area identifier in an IPv4 address format
- ← **routerId** Router identifier of the originating router
- **virtNbrRestartHelperStatus** Neighbor restart helper status:
 - OSPF6_API_RESTART_HELPER
 - Graceful restart helper
 - SPF6_API_RESTART_HELPER_NONE
 - Not a graceful restart helper

Returns:

0 on success, otherwise following error codes OSPF6_API_GET_ERROR

2.1.2.166 `int smi_ospf6_get_virt_nbr_state (struct smiclient_globals * azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr routerId, int * virtNbrState)`

This call gets the state of the relationship with this virtual Neighbor. `smi_ospf6_get_virt_nbr_state`

Parameters:

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

- ← *tag* OSPFv3 process tag
- ← *areaId* The Transit Area ID.
- ← *routerId* The router ID of the virtual neighbor
- *virtNbrState* The state of the relationship with this Neighbor

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_GET_ERROR

2.1.2.167 int smi_ospf6_graceful_restart_set (struct smiclient_globals * azg, u_int32_t vrId, int gracePeriod, int ospf6RestartGracefulReason)

This function sets the grace period(in seconds) for restarting the router. smi_ospf6_graceful_restart_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *gracePeriod* Grace period in seconds <1-1800>
- ← *reason* Reason for restart <0-2>
 - 0-OSPF6_RESTART_REASON_UNKNOWN
 - 1-OSPF6_RESTART_REASON_SOFTWARE
 - 2-OSPF6_RESTART_REASON_UPGRADE

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_GRACE_PERIOD_INVALID
 OSPF6_API_SET_ERR_INVALID_REASON
 OSPF6_API_SET_ERR_NO_PROCESS_EXIST

2.1.2.168 int smi_ospf6_graceful_restart_unset (struct smiclient_globals * azg, u_int32_t vrId)

This function sets the grace period(in seconds) for restarting the router. smi_ospf6_graceful_restart_unset

Parameters:

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

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST

2.1.2.169 `int smi_ospf6_if_cost_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_int32_t ospf6IfCost, int instanceId)`

This function sets the current interface output ospf6IfCost. The configuration is stored regardless of whether or not a real interface exists. `smi_ospf6_if_cost_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>
- ← *ospf6IfCost* Interface output ospf6IfCost <1-65535>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_COST_INVALID

2.1.2.170 `int smi_ospf6_if_cost_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, int instanceId)`

This function resets the configured output ospf6IfCost on the specified interface to the default value: 10. `smi_ospf6_if_cost_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.171 `int smi_ospf6_if_dead_interval_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_int32_t deadInterval, int instanceId)`

This function set the interval during which no hello packets are received and after which a neighbor is declared dead. Dead-interval is advertised in the Hello packets. When receiving Hello packets, OSPF router compares dead-interval in a receiving packet and the dead-interval configured on the receiving interface. `smi_ospf6_if_dead_interval_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>
- ← *deadInterval* dead interval in seconds <1-65535>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_DEAD_INTERVAL_INVALID

2.1.2.172 int smi_ospf6_if_dead_interval_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, int instanceId)

This function resets the Router Dead Interval of the specified interface to the default value: 40 seconds. smi_ospf6_if_dead_interval_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.173 int smi_ospf6_if_disable_all_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)

This function configures the OSPFv3 routers interconnecting to non-broadcast networks. smi_ospf6_if_disable_all_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

Returns:

0 on success, otherwise following error codes

2.1.2.174 `int smi_ospf6_if_disable_all_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

This function removes the OSPFv3 routers configuration: interconnecting to non-broadcast networks. `smi_ospf6_if_disable_all_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

Returns:

- 0 on success, otherwise following error codes

2.1.2.175 `int smi_ospf6_if_hello_interval_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_int32_t helloInterval, int instanceId)`

This function configures Hello Interval on the specified interface. `smi_ospf6_if_hello_interval_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>
- ← *helloInterval* Hello interval in seconds <1-65535>

Returns:

- 0 on success, otherwise following error codes `OSPF6_API_SET_ERR_VR_NOT_EXIST`
`OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID`
`OSPF6_API_SET_ERR_IF_HELLO_INTERVAL_INVALID`

2.1.2.176 `int smi_ospf6_if_hello_interval_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, int instanceId)`

This function resets the configured Hello interval on a specified interface to the default value: 10 seconds. `smi_ospf6_if_hello_interval_unset`

Parameters:

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

- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.177 `int smi_ospf6_if_ipv6_router_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr areaId, int ospf6AreaFormat, char * ospf6ProcessTag, int instanceId)`

This function enables OSPF routing on the specified interface. `smi_ospf6_if_ipv6_router_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name.
- ← *areaId* OSPFv3 area ID in an IPv4 address format .
- ← *format* Area ID format: Area ID format in IP Address
Area ID format as decimal
- ← *ospf6ProcessTag* OSPFv3 process tag
- ← *instanceId* Interface instance ID <0-255>.

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_INSTANCE_RANGE_INVALID
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_CONFIG
 OSPF6_API_SET_ERR_IF_OWNED_BY_OTHER_AREA
 OSPF6_API_SET_ERR_IF_OWNED_BY_OTHER_PROCESS
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_MISMATCH
 OSPF6_API_SET_ERR_IF_NO_LINKLOCAL_ADDRESS

2.1.2.178 `int smi_ospf6_if_ipv6_router_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr areaId, char * ospf6ProcessTag, int instanceId)`

This function disables OSPF routing on a specified interface. `smi_ospf6_if_ipv6_router_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *ospf6ProcessTag* OSPFv3 process tag
- ← *instanceId* Interface instance ID <0-255>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_-NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 OSPF6_API_SET_ERR_IF_OWNED_BY_OTHER_AREA
 OSPF6_API_SET_ERR_IF_OWNED_BY_OTHER_PROCESS
 OSPF6_API_SET_ERR_IF_NOT_EXIST

2.1.2.179 `int smi_ospf6_if_link_lsa_suppression_set (struct smiclient_globals *
 azg, u_int32_t vrId, char * ifName, int ospf6LsaSuppressionValue,
 int instanceId)`

This function configures the OSPFv3 routers interconnecting to non-broadcast networks. `smi_ospf6_if_neighbor_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>
- ← *addr* Neighbor address in an IPv6 format

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_-NOT_EXIST
 OSPF6_API_SET_ERR_INVALID_VALUE

`smi_ospf6_if_link_lsa_suppression_set`

Function to initialize LinkLSASuppression interface configuration parameter to the user specified value.

Parameters:

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

- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>
- ← *value* Whether to enable link LSA suppression: PAL_TRUE Enable link LSA suppression
PAL_FALSE Disable link LSA suppression

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.180 int smi_ospf6_if_mtu_ignore_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, int instanceId)

This function sets ospfv3 process not to check mtu size during database description exchange. smi_ospf6_if_mtu_ignore_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id; for a non-virtual-router implementation, specify 0
- ← *ifName* The interface name
- ← *instanceId* Interface instance ID <0-255>

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST

2.1.2.181 int smi_ospf6_if_mtu_ignore_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, int instanceId)

This function unconfigures the ospfv3 process mtu ignorance of during database description exchange. smi_ospf6_if_mtu_ignore_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface.
- ← *instanceId* Interface instance ID <0-255>

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST

**2.1.2.182 int smi_ospf6_if_network_set (struct smiclient_globals * azg,
 u_int32_t vrId, char * ifName, int ospf6IfType, int instanceId)**

This function sets the network type on the specified interface. smi_ospf6_if_network_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>
- ← *type* The network type. One of the following type OSPF_IFTYPE_POINTOPOINT
 OSPF_IFTYPE_BROADCAST
 OSPF_IFTYPE_NBMA
 OSPF_IFTYPE_POINTOMULTIPOINT
 OSPF_IFTYPE_POINTOMULTIPOINT_NBMA

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_NETWORK_TYPE_INVALID

**2.1.2.183 int smi_ospf6_if_network_unset (struct smiclient_globals * azg,
 u_int32_t vrId, char * ifName, int instanceId)**

This function removes the network type on the specified interface. smi_ospf6_if_network_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.184 `int smi_ospf6_if_priority_set (struct smiclient_globals * azg,
u_int32_t vrId, char * ifName, int ospf6IfPriority, int instanceId)`

This function configures router priority on the specified interface. smi_ospf6_if_priority_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>
- ← *priority* Router priority of the interface <0-255>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
OSPF6_API_SET_ERR_IF_PRIORITY_INVALID

2.1.2.185 `int smi_ospf6_if_priority_unset (struct smiclient_globals * azg,
u_int32_t vrId, char * ifName, int instanceId)`

This function sets the router priority of the specified interface to default value 0. smi_ospf6_if_priority_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.186 `int smi_ospf6_if_retransmit_interval_set (struct smiclient_globals *
azg, u_int32_t vrId, char * ifName, u_int32_t retransmitInterval, int
instanceId)`

This function sets the interval between retransmission of Link State Update packets for adjacencies belonging to the interface. smi_ospf6_if_retransmit_interval_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>
- ← *retransmitInterval* Retransmit Interval in seconds <1-65535>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_-NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_RETRANSMIT_INTERVAL_INVALID

2.1.2.187 int smi_ospf6_if_retransmit_interval_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, int instanceId)

This function resets the interval between retransmission of Link State Update packets for adjacencies belonging to the interface to the default value: 5 seconds. smi_ospf6_if_retransmit_interval_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_-NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.188 int smi_ospf6_if_te_metric_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_int32_t ospf6TeMetric, int instanceId)

This function sets the specified metric for OSPFv3 traffic engineering on an interface. smi_ospf6_if_te_metric_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0

- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>
- ← *metric* Interface TE metric <1-65535>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_METRIC_INVALID

2.1.2.189 int smi_ospf6_if_te_metric_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, int instanceId)

This function unsets a TE metric for OSPFv3 traffic engineering on an interface. smi_ospf6_if_te_metric_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

2.1.2.190 int smi_ospf6_if_transmit_delay_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_int32_t transmitDelay, int instanceId)

This function sets the estimated time it takes to transmit a Link State Update packet over the interface. smi_ospf6_if_transmit_delay_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *instanceId* Interface instance ID <0-255>
- ← *transmitDelay* Transmit Delay in seconds <1-65535>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_TRANSMIT_DELAY_INVALID

**2.1.2.191 int smi_ospf6_if_transmit_delay_unset (struct smiclient_globals *
 azg, u_int32_t vrId, char * ifName, int instanceId)**

This function resets the configured OSPF transmit delay of the specified interface to the default value: 1. smi_ospf6_if_transmit_delay_unset

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
 ← *ifName* Interface name
 ← *instanceId* Interface instance ID <0-255>

Returns:

0 on success, otherwise following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

**2.1.2.192 int smi_ospf6_ipv6_ospf_display_route_single_line_set (struct
 smiclient_globals * azg, u_int32_t vrId)**

This function displays IPv6 routing table for OSPFv3 in a single line. smi_ospf6_ipv6_ospf_display_route_single_line_set

Parameters:

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

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST

**2.1.2.193 int smi_ospf6_ipv6_ospf_display_route_single_line_unset (struct
 smiclient_globals * azg, u_int32_t vrId)**

This function displays IPv6 routing table for OSPFv3 in multiple lines. smi_ospf6_ipv6_ospf_display_route_single_line_unset

Parameters:

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

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST

2.1.2.194 `int smi_ospf6_log_adjacency_changes_set (struct smiclient_globals
 * azg, u_int32_t vrId, char * ospf6ProcessTag, u_int16_t
ospf6logAdjacencyConfig)`

This function configures the log the adjacency changes. `smi_ospf6_log_adjacency_changes_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *config* adjacency details/brief configuration
 - detail - 1 << 9
 - brief - 1 << 8

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.195 `int smi_ospf6_log_adjacency_changes_unset (struct smiclient_globals
 * azg, u_int32_t vrId, char * ospf6ProcessTag, u_int16_t
ospf6logAdjacencyConfig)`

This function unconfigures the configured logging of adjacency changes. `smi_ospf6_log_adjacency_changes_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *config* adjacency details/brief configuration
 - detail - 1 << 9
 - brief - 1 << 8

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.196 int smi_ospf6_max_concurrent_dd_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, u_int16_t maxDBDescriptor)

This function sets the maximum number of concurrently processed database descriptors. smi_ospf6_max_concurrent_dd_set

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *maxDBDescriptor* The maximum number of database descriptor processes numeric <1-65535>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.197 int smi_ospf6_max_concurrent_dd_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)

This function sets the maximum concurrent database descriptors to its default value as 5. smi_ospf6_max_concurrent_dd_unset

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.198 `int smi_ospf6_max_unuse_lsa_set (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag, u_int32_t maxUnuseLsa)`

This function sets the maximum number of unused Link-state advertisement (LSA) packets. `smi_ospf6_max_unuse_lsa_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *maxUnuseLsa* Maximum no LSA packets numeric <0-65535>

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.199 `int smi_ospf6_max_unuse_lsa_unset (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag)`

This function unsets the maximum number of unused Link-state advertisement (LSA) packets to default as (200-OSPF6_LSA_UNUSE_MAX_DEFAULT). `smi_ospf6_max_unuse_lsa_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.200 `int smi_ospf6_max_unuse_packet_set (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag, u_int32_t maxUnusePkt)`

This function sets the maximum number of unused OSPFv3 packets. `smi_ospf6_max_unuse_packet_set`

Parameters:

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

- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *maxUnusePkt* Maximum number of unused OSPFv3 packets numeric <0-65535>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.201 **int smi_ospf6_max_unuse_packet_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)**

This function sets the maximum number of unused OSPFv3 packets to its default as (200-OSPF6_PACKET_UNUSE_MAX_DEFAULT). smi_ospf6_max_unuse_packet_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.202 **int smi_ospf6_no_debug (struct smiclient_globals * azg, u_int32_t vrId, int ospf6Debug)**

Use this function to disable debugging option for OSPFv3 ZebOS information. smi_ospf6_no_debug

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *debug* Pass debug flag as following:
 - SMI_OSPF6_DBG_ALL - Enable or disable debugging for ifsm,nsfm,lsa,nsm,events and route
 - SMI_OSPF6_DBG_IFSM - Debug OSPF Interface State Machine
 - SMI_OSPF6_DBG_IFSM_EVENTS - Debug OSPF Interface State Machine events information

SMI_OSPF6_DBG_IFSM_STATUS - Debug OSPF Interface State Machine status information
SMI_OSPF6_DBG_IFSM_TIMERS - Debug OSPF Interface State Machine timers information
SMI_OSPF6_DBG_NFSM - Debug OSPF Neighbor State Machine
SMI_OSPF6_DBG_NFSM_EVENTS - Debug OSPF Neighbor State Machine events information
SMI_OSPF6_DBG_NFSM_STATUS - Debug OSPF Neighbor State Machine status information
SMI_OSPF6_DBG_NFSM_TIMERS - Debug OSPF Neighbor State Machine timers information
SMI_OSPF6_DBG_LSA - Debug OSPF Link State Advertisement
SMI_OSPF6_DBG_LSA_FLOODING - Debug LSA flooding
SMI_OSPF6_DBG_LSA_GENERATE - Debug LSA generation
SMI_OSPF6_DBG_LSA_INSTALL - Debug LSA installation
SMI_OSPF6_DBG_LSA_MAXAGE - Debug the maximum age processing
SMI_OSPF6_DBG_LSA_REFRESH - Debug LSA refresh
SMI_OSPF6_DBG_NSM - Debug OSPF NSM information
SMI_OSPF6_DBG_NSM_INTERFACE - Debug NSM interface information
SMI_OSPF6_DBG_NSM_REDISTRIBUTE - Debug NSM redistribute information
SMI_OSPF6_DBG_RIB - Debug OSPF RIB information
SMI_OSPF6_DBG_RIB_INTERFACE - Debug RIB interface information
SMI_OSPF6_DBG_RIB_REDISTRIBUTE - Debug RIB redistribute information
SMI_OSPF6_DBG_EVENT - Debug OSPF event trouble shooting
SMI_OSPF6_DBG_EVENT_ABR - Debug OSPF ABR events
SMI_OSPF6_DBG_EVENT_ASBR - Debug OSPF ASBR events
SMI_OSPF6_DBG_EVENT_LSA - Debug OSPF LSA events
SMI_OSPF6_DBG_EVENT_NSSA - Debug OSPF NSSA events
SMI_OSPF6_DBG_EVENT_OS - Debug OSPF OS events
SMI_OSPF6_DBG_EVENT_ROUTER - Debug OSPF ROUTER events
SMI_OSPF6_DBG_EVENT_VLINK - Debug OSPF VLINK events
SMI_OSPF6_DBG_ROUTE_CALC - Debug route calculation
SMI_OSPF6_DBG_ROUTE_CALC_ASE - Debug OSPF external route calculation
SMI_OSPF6_DBG_ROUTE_CALC_IA - Debug OSPF inter-area route calculation
SMI_OSPF6_DBG_ROUTE_INSTALL - Debug OSPF route installation
SMI_OSPF6_DBG_ROUTE_CALC_SPF - Debug OSPF SPF calculation
SMI_OSPF6_DBG_BFD - Debug Bidirectional Forwarding Detection

Returns:

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

SMI_ERROR

2.1.2.203 `int smi_ospf6_passive_if_default_set (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag)`

This function sets all the interfaces into passive mode. `smi_ospf6_passive_if_default_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.204 `int smi_ospf6_passive_if_default_unset (struct smiclient_globals *
azg, u_int32_t vrId, char * ospf6ProcessTag)`

This function removes all the interfaces from passive and no passive interface list. `smi_ospf6_passive_if_default_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.205 `int smi_ospf6_passive_if_set (struct smiclient_globals * azg, u_int32_t
vrId, char * ifName, char * ospf6ProcessTag)`

This function configures the specified interface into passive mode. `smi_ospf6_passive_if_set`

Parameters:

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

- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *ifName* Interface name string

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_PASSIVE_IF_ENTRY_EXIST

2.1.2.206 int smi_ospf6_passive_if_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, char * ospf6ProcessTag)

This function unsets the passive interface configuration. smi_ospf6_passive_if_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *ifName* Interface name string

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_PASSIVE_IF_ENTRY_NOT_EXIST
 OSPF6_API_SET_ERR_NO_PASSIVE_IF_ENTRY_EXIST

2.1.2.207 int smi_ospf6_process_shutdown_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)

This function configures the ospfv3 process to shutdown. smi_ospf6_process_shutdown_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.208 `int smi_ospf6_process_shutdown_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)`

This function unconfigures the ospfv3 process shutdown configuration. `smi_ospf6_process_shutdown_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.209 `int smi_ospf6_redist_default_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistDefaultOrigin)`

This function sets default redistributed protocol table as from default information originate. `smi_ospf6_redist_default_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *origin* Default information originate as OSPF6_DEFAULT_ORIGINATE_ALWAYS

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_DEFAULT_ORIGIN_INVALID

2.1.2.210 `int smi_ospf6_redist_default_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)`

This function unsets the default redistributed protocol table as from default information originate. `smi_ospf6_redist_default_unset`

Parameters:

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

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

← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes

OSPF6_API_SET_ERR_VR_NOT_EXIST

OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.211 `int smi_ospf6_redistribute_metric_set_by_ospf6_src_tag_sdkapi`
 (struct smiclient_globals * *azg*, u_int32_t *vrId*, char *
ospf6ProcessTag, int *ospf6RedistProtocol*, char * *sourceOspf6Tag*,
 u_int32_t *ospf6RedistMetricValue*)

This function sets the metric for a redistributed route. smi_ospf6_redistribute_metric_set_by_ospf6_src_tag_sdkapi

Parameters:

← *azg* Pointer to the SMI client global structure

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

← *ospf6ProcessTag* OSPFv3 Process tag

← *proto* Redistributed protocol type <1-9>

1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT

3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP

5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF

7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP

9-IPI_ROUTE_ISIS

← *sourceOspf6Tag* OSPFv3 process tag or OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance

← *metricValue* Metric value for the external route <0-16777214>

Returns:

0 on success, otherwise one of the following error codes

OSPF6_API_SET_ERR_VR_NOT_EXIST

OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID

OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

OSPF6_API_SET_ERR_METRIC_INVALID

2.1.2.212 `int smi_ospf6_redistribute_metric_set_by_ospf_src_tag_sdkapi`
 (struct smiclient_globals * *azg*, u_int32_t *vrId*, char *
ospf6ProcessTag, int *ospf6RedistProtocol*, int *sourceOspfTag*,
 u_int32_t *ospf6RedistMetricValue*)

This function sets the metric for a redistributed route. smi_ospf6_redistribute_metric_set_by_ospf_src_tag_sdkapi

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *sourceOspfTag* OSPFv3 process tag or OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance
- ← *metricValue* Metric value for the external route <0-16777214>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID
 OSPF6_API_SET_ERR_METRIC_INVALID

2.1.2.213 `int smi_ospf6_redistribute_metric_type_set_by_ospf6_src_tag_sdkapi`
 (struct smiclient_globals * *azg*, u_int32_t *vrId*, char *
ospf6ProcessTag, int *ospf6RedistProtocol*, char * *sourceOspf6Tag*,
 u_char *ospf6RedistMetricType*)

This function sets the metric type for a redistributed route. smi_ospf6_redistribute_metric_type_set_by_ospf6_src_tag_sdkapi

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *sourceOspf6tag* OSPFv3 process tag used to redistribute from one OSPF instance to another OSPF instance

← *metricType* Metric type <1-2>
 1-EXTERNAL_METRIC_TYPE_1
 2-EXTERNAL_METRIC_TYPE_2

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_METRIC_TYPE_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.214 `int smi_ospf6_redistribute_metric_type_set_by_ospf_src_tag_sdkapi`
 (struct smiclient_globals * *azg*, u_int32_t *vrId*, char *
ospf6ProcessTag, int *ospf6RedistProtocol*, int *sourceOspfTag*, u_char
ospf6RedistMetricType)

This function sets the metric type for a redistributed route. `smi_ospf6_redistribute_metric_type_set_by_ospf_src_tag_sdkapi`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *proto* Redistributed protocol type <1-9>
 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 9-IPI_ROUTE_ISIS
 ← *sourceOspfTag* OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance
 ← *metricType* Metric type <1-2>
 1-EXTERNAL_METRIC_TYPE_1
 2-EXTERNAL_METRIC_TYPE_2

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_METRIC_TYPE_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.215 `int smi_ospf6_redistribute_metric_type_unset_by_ospf6_src_tag -
sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char *
ospf6ProcessTag, int ospf6RedistProtocol, char * sourceOspf6Tag)`

This function sets the metric type of a redistributed route to its default value 2. `smi_ospf6_redistribute_metric_type_unset_by_ospf6_src_tag_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *sourceOspf6Tag* OSPFv3 process tag used to redistribute from one OSPF instance to another OSPF instance

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
- OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.216 `int smi_ospf6_redistribute_metric_type_unset_by_ospf_src_tag -
sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char *
ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)`

This function sets the metric type of a redistributed route to its default value 2. `smi_ospf6_redistribute_metric_type_unset_by_ospf_src_tag_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS

← *sourceOspfTag* OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.217 `int smi_ospf6_redistribute_metric_unset_by_ospf6_src_tag_sdkapi`
 (struct smiclient_globals * *azg*, u_int32_t *vrId*, char *
ospf6ProcessTag, int *ospf6RedistProtocol*, char * *sourceOspfTag*)

This function sets the metric of a redistributed route to its default value 16777215.
 smi_ospf6_redistribute_metric_unset_by_ospf6_src_tag_sdkapi

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *proto* Redistributed protocol type <1-9>
 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 9-IPI_ROUTE_ISIS
 ← *sourceOspfTag* OSPFv3 process tag or OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.218 `int smi_ospf6_redistribute_metric_unset_by_ospf_src_tag_sdkapi`
 (struct smiclient_globals * *azg*, u_int32_t *vrId*, char *
ospf6ProcessTag, int *ospf6RedistProtocol*, int *sourceOspfTag*)

This function sets the metric of a redistributed route to its default value 16777215.
 smi_ospf6_redistribute_metric_unset_by_ospf_src_tag_sdkapi

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrId** Virtual Router ID numeric <0-255>
- ← **ospf6ProcessTag** OSPFv3 Process tag
- ← **proto** Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← **sourceOspfTag** OSPFv3 process tag or OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
- OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.219 `int smi_ospf6_redistribute_route_tag_set_by_ospf6_src_tag_sdkapi`
 (struct smiclient_globals * *azg*, u_int32_t *vr_id*, char *
ospf6ProcessTag, int *ospf6RedistProtocol*, char * *sourceOspfTag*,
 u_int32_t *ospf6RedistTagVal*)

This function sets the metric for a redistributed route. `smi_ospf6_redistribute_route_tag_set_by_ospf6_src_tag_sdkapi`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID numeric <0-255>
- ← **tag** OSPFv3 Process tag
- ← **proto** Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← **source_ospf6tag** OSPFv3 process tag or OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance
- ← **mvalue** Metric value for the external route <0-4294967295>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID
 OSPF6_API_SET_ERR_METRIC_INVALID

2.1.2.220 `int smi_ospf6_redistribute_route_tag_set_by_ospf_src_tag_sdkapi`
 (struct smiclient_globals * *azg*, u_int32_t *vrId*, char *
ospf6ProcessTag, int *ospfRedistProtocol*, int *sourceOspfTag*,
 u_int32_t *ospfRedistTagVal*)

This function sets the metric for a redistributed route. smi_ospf6_redistribute_route_tag_set_by_ospf_src_tag_sdkapi

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *tag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *source_ospftag* OSPFv3 process tag or OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance
- ← *tvalue* Metric value for the external route <0-4294967295>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID
 OSPF6_API_SET_ERR_METRIC_INVALID

2.1.2.221 `int smi_ospf6_redistribute_route_tag_unset_by_ospf6_src_tag_sdkapi`
 (struct smiclient_globals * *azg*, u_int32_t *vrId*, char *
ospf6ProcessTag, int *ospf6RedistProtocol*, char * *sourceOspf6Tag*)

This function sets the metric of a redistributed route to its default value 16777215. smi_ospf6_redistribute_route_tag_unset_by_ospf6_src_tag_sdkapi

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID numeric <0-255>
- ← **tag** OSPFv3 Process tag
- ← **proto** Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← **source_ospf6tag** OSPFv3 process tag or OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.222 int smi_ospf6_redistribute_route_tag_unset_by_ospf_src_tag_sdkapi
 (struct smiclient_globals * **azg**, u_int32_t **vrId**, char *
ospf6ProcessTag, int **ospfRedistProtocol**, int **sourceOspfTag**)

This function sets the metric of a redistributed route to its default value 16777215.
 smi_ospf6_redistribute_route_tag_unset_by_ospf_src_tag_sdkapi

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr_id** Virtual Router ID numeric <0-255>
- ← **tag** OSPFv3 Process tag
- ← **proto** Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← **source_ospftag** OSPFv3 process tag or OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance

Returns:

0 on success, otherwise one of the following error codes

```

OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

```

2.1.2.223 `int smi_ospf6_redistribute_set_by_ospf6_src_tag_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistProtocol, char * sourceOspf6Tag)`

This function redistributes routes from the specified protocol. `smi_ospf6_redistribute_set_by_ospf6_src_tag_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *sourceOspf6tag* OSPFv3 process tag used to redistribute from one OSPF instance to another OSPF instance

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_SELF_TAG

2.1.2.224 `int smi_ospf6_redistribute_set_by_ospf_src_tag_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)`

This function redistributes routes from the specified protocol. `smi_ospf6_redistribute_set_by_ospf_src_tag_sdkapi`

Parameters:

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

- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *sourceOspfTag* OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_SELF_TAG

2.1.2.225 `int smi_ospf6_redistribute_unset_by_ospf6_src_tag_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistProtocol, char * sourceOspf6Tag)`

This function stops redistributing routes from the specified protocol. `smi_ospf6_redistribute_unset_by_ospf6_src_tag_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *sourceOspf6tag* OSPFv3 process tag used to redistribute from one OSPF instance to another OSPF instance

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.226 `int smi_ospf6_redistribute_unset_by_ospf_src_tag_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)`

This function stops redistributing routes from the specified protocol. `smi_ospf6_redistribute_unset_by_ospf_src_tag_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *sourceOspfTag* OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
- OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.227 `int smi_ospf6_restart_graceful_set (struct smiclient_globals * azg, u_int32_t vrId, u_int16_t gracePeriod, int ospf6RestartGracefulReason)`

This function causes a graceful restart of the router. After this command executes, the router immediately shuts down and notifies NSM that the shut down was graceful. `smi_ospf6_restart_graceful`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *gracePeriod* Grace period in seconds <1-1800>
- ← *reason* Reason for restart <0-2>
 - 0-OSPF6_RESTART_REASON_UNKNOWN
 - 1-OSPF6_RESTART_REASON_SOFTWARE
 - 2-OSPF6_RESTART_REASON_UPGRADE

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_GRACE_PERIOD_INVALID
 OSPF6_API_SET_ERR_INVALID_REASON
 OSPF6_API_SET_ERR_NO_PROCESS_EXIST

2.1.2.228 int smi_ospf6_restart_helper_grace_period_set (struct smiclient_globals * azg, u_int32_t vrId, int gracePeriod)

This function sets the configured maximum grace period for a neighbor to act as helper.
 smi_ospf6_restart_helper_grace_period_set

Parameters:

← **azg** Pointer to the SMI client global structure
 ← **vrId** Virtual Router ID numeric <0-255>
 ← **gracePeriod** Grace period numeric <1-1800>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_GRACE_PERIOD_INVALID

2.1.2.229 int smi_ospf6_restart_helper_grace_period_unset (struct smiclient_globals * azg, u_int32_t vrId)

This function reverts the configured maximum grace period for a neighbor to act as helper to the default value as 1. smi_ospf6_restart_helper_grace_period_unset

Parameters:

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

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST

2.1.2.230 int smi_ospf6_restart_helper_never_router_id_set (struct smiclient_globals * azg, u_int32_t vrId, struct pal_in4_addr routerId)

This function adds the specified router ID to the Never-Router ID list. smi_ospf6_restart_helper_never_router_id_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *routerId* Router ID in an IPv4 format

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_ROUTER_ID_ALREADY_CONFIGURED
OSPF6_API_SET_ERR_MALLOC_FAIL_FOR_ROUTERID

2.1.2.231 int smi_ospf6_restart_helper_never_router_id_unset (struct smiclient_globals * *azg*, u_int32_t *vrId*, struct pal_in4_addr *routerId_ptr*)

This function deletes the specified router ID to the Never-Router ID list. smi_ospf6_restart_helper_never_router_id_unset

Parameters:

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

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_EMPTY_NEVER_RTR_ID
OSPF6_API_SET_ERR_NEVER_RTR_ID_NOT_EXIST

2.1.2.232 int smi_ospf6_restart_helper_never_router_unset_all (struct smiclient_globals * *azg*, u_int32_t *vrId*)

This function deletes entire Never-Router ID list. smi_ospf6_restart_helper_never_router_unset_all

Parameters:

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

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_EMPTY_NEVER_RTR_ID

2.1.2.233 `int smi_ospf6_restart_helper_policy_set (struct smiclient_globals * azg, u_int32_t vrId, int ospf6HelperPolicy)`

This function configures the specified helper policy. `smi_ospf6_restart_helper_policy_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *policy* Helper policy (2 | 4 | 8)
 - 1 << 1-OSPF6_RESTART_HELPER_NEVER
 - 1 << 2-OSPF6_RESTART_HELPER_ONLY_RELOAD
 - 1 << 3-OSPF6_RESTART_HELPER_ONLY_UPGRADE

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_INVALID_HELPER_POLICY

2.1.2.234 `int smi_ospf6_restart_helper_policy_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospf6HelperPolicy)`

This function resets the configuration to default. `smi_ospf6_restart_helper_policy_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *policy* Helper policy (2 | 4 | 8)
 - 1 << 1-OSPF6_RESTART_HELPER_NEVER
 - 1 << 2-OSPF6_RESTART_HELPER_ONLY_RELOAD
 - 1 << 3-OSPF6_RESTART_HELPER_ONLY_UPGRADE

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_INVALID_HELPER_POLICY

2.1.2.235 `int smi_ospf6_restart_helper_policy_unset_all (struct smiclient_globals * azg, u_int32_t vrId)`

This function resets all configured policies. `smi_ospf6_restart_helper_policy_unset_all`

Parameters:

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

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST

2.1.2.236 `int smi_ospf6_routemap_default_set (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag, char * routeMapName)`

This function sets default route map from a redistributed protocol. `smi_ospf6_routemap_default_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *routeMapName* Route-map name

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
OSPF6_API_SET_MALLOC_ERR

2.1.2.237 `int smi_ospf6_routemap_default_unset (struct smiclient_globals *
azg, u_int32_t vrId, char * ospf6ProcessTag)`

This function unsets default route map from a redistributed protocol. `smi_ospf6_routemap_default_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.238 `int smi_ospf6_routemap_set_by_ospf6_src_tag_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistProtocol, char * sourceOspf6Tag, char * routeMapName)`

This function assigns a route-map to a redistributed protocol. `smi_ospf6_routemap_set_by_ospf6_src_tag_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *sourceOspf6Tag* OSPFv3 process tag used to redistribute from one OSPF instance to another OSPF instance
- ← *routeMapName* Route-map name

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.239 `int smi_ospf6_routemap_set_by_ospf_src_tag_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag, char * routeMapName)`

This function assigns a route-map to a redistributed protocol. `smi_ospf6_routemap_set_by_ospf_src_tag_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS

- ← *sourceOspfTag* OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance
- ← *routeMapName* Route-map name

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.240 int smi_ospf6_routemap_unset_by_ospf6_src_tag_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistProtocol, char * sourceOspf6Tag)

This function removes a route map from a redistributed protocol. smi_ospf6_routemap_unset_by_ospf6_src_tag_sdkapi

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *sourceOspf6Tag* OSPFv3 process tag used to redistribute from one OSPF instance to another OSPF instance

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.241 int smi_ospf6_routemap_unset_by_ospf_src_tag_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int ospf6RedistProtocol, int sourceOspfTag)

This function removes a route map from a redistributed protocol. smi_ospf6_routemap_unset_by_ospf_src_tag_sdkapi

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *proto* Redistributed protocol type <1-9>
 - 1-IPI_ROUTE_KERNEL, 2-IPI_ROUTE_CONNECT
 - 3-IPI_ROUTE_STATIC, 4-IPI_ROUTE_RIP
 - 5-IPI_ROUTE_RIPNG, 6-IPI_ROUTE_OSPF
 - 7-IPI_ROUTE_OSPF6, 8-IPI_ROUTE_BGP
 - 9-IPI_ROUTE_ISIS
- ← *sourceOspfTag* OSPF process identifier used to redistribute from one OSPF instance to another OSPF instance

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_INVALID
 OSPF6_API_SET_ERR_REDISTRIBUTE_PROTO_TAG_INVALID

2.1.2.242 `int smi_ospf6_router_id_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr routerId)`

This function configures the OSPFv3 router ID. `smi_ospf6_routerId_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 process tag
- ← *routerId* Router ID in IPv4 format

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_routerId_INVALID
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.243 `int smi_ospf6_router_id_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)`

This function forces the OSPFv3 to stop routing functionality. `smi_ospf6_routerId_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 process tag

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.244 int smi_ospf6_router_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)

This function initiates the OSPFv3 router instance. smi_ospf6_router_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_CANT_INITIATE
OSPF6_API_SET_ERR_PID_CANT_ASSIGN

2.1.2.245 int smi_ospf6_router_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag)

This function removes the OSPFv3 router instance. smi_ospf6_router_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.246 `int smi_ospf6_set_area_nssa_trans_role (struct smiclient_globals *
azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, u_int32_t
ospf6AreaNssaTransRole)`

This function sets an NSSA border router's policy to perform NSSA(Not-so-Stubby Area) translation of NSSA-LSAs into(Autonomous System) AS-External-LSAs. smi_ospf6_set_area_nssa_trans_role

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 Process tag
- ← **areaId** OSPFv3 area ID in an IPv4 address format
- ← **role** NSSA(Not-so-Stubby Area) role
 - 1-OSPF_NSSA_TRANSLATE_ALWAYS
 - 2-OSPF_NSSA_TRANSLATE_CANDIDATE

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_IS_BACKBONE
 OSPF6_API_SET_ERR_AREA_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_NOT_NSSA
 OSPF6_API_SET_ERROR

2.1.2.247 `int smi_ospf6_set_area_stub_metric_type (struct smiclient_globals *
azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int
ospf6StubMetricType)`

This function sets the type of metric advertised as a default route. smi_ospf6_set_area_stub_metric_type

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **tag** OSPFv3 Process tag
- ← **areaId** OSPFv3 area ID in an IPv4 address format
- ← **metricType** Metric type
 - 1-EXTERNAL_METRIC_TYPE_1
 - 2-EXTERNAL_METRIC_TYPE_2

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST

```

OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
OSPF6_API_SET_ERR_AREA_IS_BACKBONE
OSPF6_API_SET_ERR_METRIC_TYPE_INVALID
OSPF6_API_SET_ERR_AREA_NOT_EXIST
OSPF6_API_SET_ERR_AREA_NOT_NSSA
OSPF6_API_SET_ERROR

```

2.1.2.248 `int smi_ospf6_set_area_te_enabled (struct smiclient_globals *
azg, char * ospf6ProcessTag, struct pal_in4_addr areaId, int
ospf6TeStatus)`

This function sets to whether or not traffic engineering is enabled in the area. smi_ospf6_set_area_te_enabled

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *areaId* OSPFv3 area ID in an IPv4 address format
- ← *status* Traffic engineering status
 - 1-OSPF6_API_STATUS_ENABLED
 - 2-OSPF6_API_STATUS_DISABLED

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERROR

2.1.2.249 `int smi_ospf6_set_asbdr_rtr_status (struct smiclient_globals * azg,
char * ospf6ProcessTag, int ospf6AsbdrRtrStatus)`

This function sets the router as an Autonomous System (AS) border router. smi_ospf6_set_asbdr_rtr_status

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 Process tag
- ← *status* Router status
 - AS border router-OSPF6_API_TRUE
 - Not an AS border router-Disabled-OSPF6_API_FALSE

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERROR

2.1.2.250 `int smi_ospf6_set_if_admin_stat (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int adminstat)`

smi_ospf6_set_if_admin_stat function sets the administrative status of the given OSPFv3 interface.

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier
- ← *adminstat* Administrative status of the OSPFv3 interface
 - Enabled-OSPF6_API_STATUS_ENABLED
 - Disabled-OSPF6_API_STATUS_DISABLED

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_SET_ERROR
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_NETWORK_TYPE_INVALID
 OSPF6_API_SET_ERR_AREA_ID_FORMAT_INVALID
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_INSTANCE_RANGE_INVALID
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_CONFIG
 OSPF6_API_SET_ERR_IF_OWNED_BY_OTHER_AREA
 OSPF6_API_SET_ERR_IF_OWNED_BY_OTHER_PROCESS
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_MISMATCH
 OSPF6_API_SET_ERR_IF_NO_LINKLOCAL_ADDRESS
 OSPF6_API_SET_ERR_IF_PARAM_NOT_CONFIGURED
 OSPF6_API_SET_ERR_IF_NOT_EXIST

2.1.2.251 `int smi_ospf6_set_if_rtr_dead_interval (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int ifRtrDeadInterval)`

This function sets the number of seconds that a router's Hello packets have not been seen on the given interface before its neighbors declare the router down. smi_ospf6_set_if_rtr_dead_interval

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *tag* OSPFv3 process tag
- ← *ifindex* Interface index
- ← *instanceId* Instance identifier

← *ifRtrDeadInterval* Value of the router-dead interval to be stored (seconds):
<1-65335>

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_SET_ERROR
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_IF_DEAD_INTERVAL_INVALID

2.1.2.252 int smi_ospf6_set_if_type (struct smiclient_globals * azg, char * ospf6ProcessTag, int ifindex, int instanceId, int ospf6IfType)

This function sets the interface type of the given OSPFv3 interface. smi_ospf6_set_if_type

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *tag* OSPFv3 process tag
 ← *ifindex* Interface index
 ← *instanceId* Instance identifier
 ← *type* Interface types
 Broadcast-OSPF_IFTYPE_BROADCAST
 Non-broadcast multi-access-OSPF_IFTYPE_NBMA
 Point-to-Point-OSPF_IFTYPE_POINTOPOINT
 Point-to-Multipoint-OSPF_IFTYPE_POINTOMULTIPOINT

Returns:

0 on success , otherwise the following error codes
 OSPF6_API_SET_ERROR
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_NETWORK_TYPE_INVALID

2.1.2.253 int smi_ospf6_show_border_routers (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int startIndex, int endIndex, struct list * brInfo, int(*) (struct list * brInfo) funpointer)

This function returns the list of border routers info. smi_ospf6_show_border_routers

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0

- ← **tag** OSPF process tag
- ← **startIndex** Starting index of the node of the returned list, if not specified it is considered be 0
- ← **endIndex** Ending index of the node of the returned list, if not specified it is considered be 0
- **brInfo** Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.254 `int smi_ospf6_show_database (struct smiclient_globals * azg, u_int32_t vrid, char * ospf6ProcessTag, int startIndex, int endIndex, struct list * dbInfo, int(*) (struct list * dbInfo) funpointer)`

This function returns the list of process database info. smi_ospf6_show_database

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrid** Virtual router ID; for a non-virtual-router implementation, specify 0
- ← **tag** OSPF process tag
- ← **startIndex** Starting index of the node of the returned list, if not specified it is considered be 0
- ← **endIndex** Ending index of the node of the returned list, if not specified it is considered be 0
- **dbInfo** Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.255 `int smi_ospf6_show_debug (struct smiclient_globals * azg, u_int32_t vrid, struct ospf6_debug_show * debug_show)`

Use this function to show debugging option for OSPFv3 ZebOS information. smi_ospf6_show_debug

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrid** Virtual Router ID

Returns:

struct ospf6_debug_show on success, otherwise one of the following error codes
 OSPF_API_SET_ERR_VR_NOT_EXIST
 SMI_ERROR

2.1.2.256 `int smi_ospf6_show_interface (struct smiclient_globals * azg,
u_int32_t vrId, char * ifName, int startIndex, int endIndex, struct
list * linterfaceInfo, int(*) (struct list * linterfaceInfo) funpointer)`

This function returns the list of Interface detail by interface name. smi_ospf6_show_
interface

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *linterfaceInfo* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.257 `int smi_ospf6_show_interface_brief (struct smiclient_globals * azg,
u_int32_t vrId, char * ifName, int startIndex, int endIndex, struct
list * interfaceBrief, int(*) (struct list * interfaceBrief) funpointer)`

This function returns the list of Interface brief by interface name. smi_ospf6_show_
interface_brief

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *interfaceBrief* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.258 `int smi_ospf6_show_interface_brief_by_tag (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, char * ifName, int startIndex, int endIndex, struct list * linterfaceInfo, int(*) (struct list * linterfaceInfo) funpointer)`

This function returns the list of Interface brief by process tag. `smi_ospf6_show_interface_brief_by_tag`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* OSPF process tag
- ← *ifName* Interface name
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *linterfaceInfo* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.259 `int smi_ospf6_show_interface_by_tag (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, char * ifName, int startIndex, int endIndex, struct list * linterfaceInfo, int(*) (struct list * linterfaceInfo) funpointer)`

This function returns the list of Interface detail by process tag. `smi_ospf6_show_interface_by_tag`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* OSPF process tag
- ← *ifName* Interface name
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *linterfaceInfo* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.260 `int smi_ospf6_show_neighbor_by_if_nbr_id (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, char * ospf6ProcessTag, char * nbrId, int startIndex, int endIndex, struct list * nbrList, int(*)(struct list *nbrList) funpointer)`

This function returns the list of neighbor info by interface name and neighbor id. `smi_ospf6_show_neighbor_by_if_nbr_id`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *tag* OSPF process tag
- ← *nbrId* IP address of neighbor router
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *nbrList* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.261 `int smi_ospf6_show_neighbor_by_interface_name (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, char * ospf6ProcessTag, int startIndex, int endIndex, struct list * nbrData, int(*)(struct sminbrByIfList *getinfo) funpointer)`

This function returns the list of neighbor by neighbor interface name. `smi_ospf6_show_neighbor_by_interface_name`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *nbrList* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.262 `int smi_ospf6_show_neighbor_by_nbr_id (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, char * nbrId, int startIndex, int endIndex, struct list * nbrData, int(*) (struct sminbrByIdList *getinfo) funpointer)`

This function returns the list of neighbor info by neighbor id. `smi_ospf6_show_neighbor_by_nbr_id`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrId** Virtual router ID; for a non-virtual-router implementation, specify 0
- ← **tag** OSPF process tag
- ← **nbrId** IP address of neighbor router
- ← **startIndex** Starting index of the node of the returned list, if not specified it is considered be 0
- ← **endIndex** Ending index of the node of the returned list, if not specified it is considered be 0
- **nbrList** Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.263 `int smi_ospf6_show_neighbor_detail_by_if_nbr_id (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, char * ospf6ProcessTag, char * nbrId, int startIndex, int endIndex, struct list * nbrList, int(*) (struct list *nbrList) funpointer)`

This function returns the list of neighbor detail info by interface name and neighbor id. `smi_ospf6_show_neighbor_detail_by_if_nbr_id`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrId** Virtual router ID; for a non-virtual-router implementation, specify 0
- ← **ifName** Interface name
- ← **tag** OSPF process tag
- ← **nbrId** IP address of neighbor router
- ← **startIndex** Starting index of the node of the returned list, if not specified it is considered be 0
- ← **endIndex** Ending index of the node of the returned list, if not specified it is considered be 0
- **nbrList** Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.264 `int smi_ospf6_show_neighbor_detail_by_interface_name (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, char * ospf6ProcessTag, int startIndex, int endIndex, struct list * nbrData, int(*) (struct sminbrDetailByIfList *getinfo) funpointer)`

This function returns the list of neighbor info by neighbor interface name. `smi_ospf6_show_neighbor_detail_by_interface_name`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *nbrList* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.265 `int smi_ospf6_show_neighbor_detail_by_tag (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int startIndex, int endIndex, struct list * nbrData, int(*) (struct sminbrDetailByTagList *getinfo) funpointer)`

This function returns the list of neighbor detail by tag. `smi_ospf6_show_neighbor_detail_by_tag`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* OSPF process tag
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *nbrList* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.266 `int smi_ospf6_show_neighbor_summary_by_if_nbr_id (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, char * ospf6ProcessTag, char * nbrid, int startIndex, int endIndex, struct list * nbrList, int(*) (struct list * nbrList) funpointer)`

This function returns the list of neighbor info summary. `smi_ospf6_show_neighbor_summary_by_if_nbr_id`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *tag* OSPF process tag
- ← *nbrid* IP address of neighbor router
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *nbrList* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.267 `int smi_ospf6_show_neighbor_summary_by_interface_name (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, char * ospf6ProcessTag, int startIndex, int endIndex, struct list * nbrList, int(*) (struct list * nbrList) funpointer)`

This function returns the list of neighbor info summary. `smi_ospf6_show_neighbor_summary_by_interface_name`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifName* Interface name
- ← *tag* OSPF process tag
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *nbrList* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.268 `int smi_ospf6_show_neighbor_summary_by_nbr_id (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, char * nbr_id, int startIndex, int endIndex, struct list * nbrList, int(*) (struct list * nbrList) funpointer)`

This function returns the list of neighbor info summary. `smi_ospf6_show_neighbor_summary_by_nbr_id`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* OSPF process tag
- ← *nbrid* IP address of neighbor router
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *nbrList* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.269 `int smi_ospf6_show_neighbor_summary_by_tag (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int startIndex, int endIndex, struct list * nbrList, int(*) (struct list * nbrList) funpointer)`

This function returns the list of neighbor info summary. `smi_ospf6_show_neighbor_summary_by_tag`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* OSPF process tag
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *nbrList* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.270 `int smi_ospf6_show_process (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int startIndex, int endIndex, struct list * procInfo, int(*) (struct list * procInfo) funpointer)`

This function returns the list of process info. `smi_ospf6_show_process`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* OSPF process tag
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *procInfo* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.271 `int smi_ospf6_show_route (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int startIndex, int endIndex, struct list * routeInfo, int(*) (struct list * routeInfo) funpointer)`

This function returns the list of route info. `smi_ospf6_show_route`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* OSPF process tag
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *routeInfo* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.272 `int smi_ospf6_show_route_by_prefix (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct prefix_ipv6 prefix, int startIndex, int endIndex, struct list * routeInfo, int(*) (struct list * routeInfo) funpointer)`

This function returns the list of route info by prefix. smi_ospf6_show_route_by_prefix

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* OSPF process tag
- ← *prefix* IPv6 Prefix address
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *routeInfo* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.273 `int smi_ospf6_show_route_summary (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, int startIndex, int endIndex, struct list * routeInfo, int(*) (struct list * routeInfo) funpointer)`

This function returns the list of route info summary. smi_ospf6_show_route_summary

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* OSPF process tag
- ← *startIndex* Starting index of the node of the returned list, if not specified it is considered be 0
- ← *endIndex* Ending index of the node of the returned list, if not specified it is considered be 0
- *routeInfo* Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.274 `int smi_ospf6_show_virtual_link (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag, int startIndex, int endIndex,
struct list * lvirtualLink, int(*) (struct list * lvirtualLink) funpointer)`

This function returns the list of virtual links. `smi_ospf6_show_virtual_link`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrId** Virtual router ID; for a non-virtual-router implementation, specify 0
- ← **tag** OSPF process tag
- ← **startIndex** Starting index of the node of the returned list, if not specified it is considered be 0
- ← **endIndex** Ending index of the node of the returned list, if not specified it is considered be 0
- **lvirtualLink** Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.275 `int smi_ospf6_show_virtual_link_brief (struct smiclient_globals * azg,
u_int32_t vrId, char * ospf6ProcessTag, int startIndex, int endIndex,
struct list * lvirtualLink, int(*) (struct list * lvirtualLink) funpointer)`

This function returns the list of virtual links brief info. `smi_ospf6_show_virtual_link_brief`

Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrId** Virtual router ID; for a non-virtual-router implementation, specify 0
- ← **tag** OSPF process tag
- ← **startIndex** Starting index of the node of the returned list, if not specified it is considered be 0
- ← **endIndex** Ending index of the node of the returned list, if not specified it is considered be 0
- **lvirtualLink** Returned list with all show information

Returns:

RETURN_OK on success else RETURN_ERROR

2.1.2.276 `int smi_ospf6_summary_address_ipv4_set_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)`

This function summarizes external routes with the specified address range. `smi_ospf6_summary_address_ipv6_set_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *ipv4Addr* IPv4 starting address
- ← *masklen* Mask length

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_SUMMARY_ADDRESS_EXIST
- OSPF6_API_SET_MALLOC_ERR

`smi_ospf6_summary_address_ipv4_set_sdkapi`

This function summarizes external routes with the specified address range.

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *ipv4Addr* IPv4 starting address
- ← *masklen* Mask length

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_SUMMARY_ADDRESS_EXIST
- OSPF6_API_SET_MALLOC_ERR

2.1.2.277 `int smi_ospf6_summary_address_ipv4_unset_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)`

This function removes a summary address. `smi_ospf6_summary_address_ipv6_unset_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *ipv6Addr* IPv6 starting address
- ← *masklen* Mask length

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_SUMMARY_ADDRESS_NOT_EXIST
 OSPF6_API_SET_MALLOC_ERR

smi_ospf6_summary_address_ipv4_unset_sdkapi

This function removes a summary address.

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *ipv4Addr* IPv4 starting address
- ← *masklen* Mask length

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_SUMMARY_ADDRESS_NOT_EXIST
 OSPF6_API_SET_MALLOC_ERR

2.1.2.278 `int smi_ospf6_summary_address_not_advertise_ipv4_set_sdkapi`
 (struct smiclient_globals * *azg*, u_int32_t *vrId*, char *
ospf6ProcessTag, struct pal_in4_addr *ipv4Addr*, u_char *masklen*)

This function suppresses external routes that match a specified address range. smi_ospf6_summary_address_not_advertise_ipv6_set_sdkapi

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *tag* OSPFv3 Process tag

← *ipv6Addr* IPv6 starting address

← *masklen* Mask length

Returns:

0 on success, otherwise one of the following error codes

OSPF6_API_SET_ERR_VR_NOT_EXIST

OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

OSPF6_API_SET_ERR_SUMMARY_ADDRESS_NOT_EXIST

smi_ospf6_summary_address_not_advertise_ipv4_set_sdkapi

This function suppresses external routes that match a specified address range.

Parameters:

← *azg* Pointer to the SMI client global structure

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

← *tag* OSPFv3 Process tag

← *ipv4Addr* IPv4 starting address

← *masklen* Mask length

Returns:

0 on success, otherwise one of the following error codes

OSPF6_API_SET_ERR_VR_NOT_EXIST

OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

OSPF6_API_SET_ERR_SUMMARY_ADDRESS_NOT_EXIST

2.1.2.279 int smi_ospf6_summary_address_not_advertise_ipv4_unset_sdkapi

(struct smiclient_globals * *azg*, u_int32_t *vrId*, char *

ospf6ProcessTag, struct pal_in4_addr *ipv4Addr*, u_char *masklen*)

This function stops suppressing external routes that match a specified address range.

smi_ospf6_summary_address_not_advertise_ipv4_unset_sdkapi

Parameters:

← *azg* Pointer to the SMI client global structure

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

← *tag* OSPFv3 Process tag

← *ipv4Addr* IPv4 starting address

← *masklen* Mask length

Returns:

0 on success, otherwise one of the following error codes

OSPF6_API_SET_ERR_VR_NOT_EXIST

OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

OSPF6_API_SET_ERR_SUMMARY_ADDRESS_NOT_EXIST

2.1.2.280 `int smi_ospf6_summary_address_not_advertise_ipv6_unset_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in6_addr ipv6Addr, u_char masklen)`

This function stops suppressing external routes that match a specified address range. `smi_ospf6_summary_address_not_advertise_ipv6_unset_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *tag* OSPFv3 Process tag
- ← *ipv6Addr* IPv6 starting address
- ← *masklen* Mask length

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_SUMMARY_ADDRESS_NOT_EXIST

2.1.2.281 `int smi_ospf6_summary_address_tag_ipv4_set_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen, u_int32_t routeTag)`

This function sets a `ospf6ProcessTag` value to use as a "match" value for controlling redistribution via route maps. `smi_ospf6_summary_address_tag_ipv6_set_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *ipv6Addr* IPv6 starting address
- ← *masklen* Mask length
- ← *routeTag* Route tag value

Returns:

- 0 on success, otherwise one of the following error codes
- OSPF6_API_SET_ERR_VR_NOT_EXIST
- OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
- OSPF6_API_SET_ERR_SUMMARY_ADDRESS_NOT_EXIST

`smi_ospf6_summary_address_tag_ipv4_set_sdkapi`

This function sets a tag value to use as a "match" value for controlling redistribution via route maps.

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *ospf6ProcessTag* OSPFv3 Process tag
- ← *ipv4Addr* IPv4 starting address
- ← *masklen* Mask length
- ← *routeTag* Route tag value

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_SUMMARY_ADDRESS_NOT_EXIST

2.1.2.282 `int smi_ospf6_summary_address_tag_ipv4_unset_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr ipv4Addr, u_char masklen)`

This function removes a tag value to use as a "match" value for controlling redistribution via route maps. `smi_ospf6_summary_address_tag_ipv6_unset_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *tag* OSPFv3 Process tag
- ← *ipv6Addr* IPv6 starting address
- ← *masklen* Mask length

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_SUMMARY_ADDRESS_NOT_EXIST

`smi_ospf6_summary_address_tag_ipv4_unset_sdkapi`

This function removes a tag value to use as a "match" value for controlling redistribution via route maps.

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID numeric <0-255>
- ← *tag* OSPFv3 Process tag

← *ipv4Addr* IPv4 starting address
 ← *masklen* Mask length

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_SUMMARY_ADDRESS_NOT_EXIST

2.1.2.283 `int smi_ospf6_timers_spf_set (struct smiclient_globals * azg,
 u_int32_t vrId, char * ospf6ProcessTag, u_int32_t minDelay,
 u_int32_t maxDelay)`

This function sets the minimum and maximum delay between a topology change, being either received in an LSA(Link-state advertisement) or self detected and the SP(Shortest Path First)F being run. smi_ospf6_timers_spf_set

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *minDelay* The minimum SPF hold delay time in milliseconds
 ← *maxDelay* The minimum SPF hold delay time in milliseconds

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_TIMER_VALUE_INVALID

2.1.2.284 `int smi_ospf6_timers_spf_unset (struct smiclient_globals * azg,
 u_int32_t vrId, char * ospf6ProcessTag)`

This function sets SPF minimum and maximum delay to their default values. Minimum default delay is 500 milliseconds and Maximum default delay is 50000 milliseconds. smi_ospf6_timers_spf_unset

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.1.2.285 `int smi_ospf6_vlink_dead_interval_set (struct smiclient_globals * azg,
 u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId,
 struct pal_in4_addr peerId, int virtualLinkDeadInterval)`

This function configures the Router Dead Interval of the specified virtual link. `smi_ospf6_vlink_dead_interval_set`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *areaId* OSPFv3 area ID in an IPv4 address format
 ← *peerId* Neighbor ID
 ← *vDeadInterval* Router dead interval numeric <1-65535>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_IF_DEAD_INTERVAL_INVALID
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.1.2.286 `int smi_ospf6_vlink_dead_interval_unset (struct smiclient_globals *
azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr
areaId, struct pal_in4_addr peerId)`

This function unsets the configured Router Dead Interval of the specified virtual link. `smi_ospf6_vlink_dead_interval_unset`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *areaId* OSPFv3 area ID in an IPv4 address format
 ← *peerId* Neighbor ID

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.1.2.287 `int smi_ospf6_vlink_hello_interval_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkHelloInterval)`

This function configures Hello Interval on the specified virtual link. `smi_ospf6_vlink_hello_interval_set`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *areaId* OSPFv3 area ID in an IPv4 address format
 ← *peerId* Neighbor ID
 ← *vHelloInterval* The interval (seconds) before the router sends the hello packet numeric <1-65535>.

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_IF_HELLO_INTERVAL_INVALID
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.1.2.288 `int smi_ospf6_vlink_hello_interval_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)`

This function resets the Hello Interval on a specified virtual link to the default value as 10 seconds. `smi_ospf6_vlink_hello_interval_unset`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *areaId* OSPFv3 area ID in an IPv4 address format
 ← *peerId* Neighbor ID

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_IF_HELLO_INTERVAL_INVALID
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.1.2.289 `int smi_ospf6_vlink_instance_id_set (struct smiclient_globals * azg,
 u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId,
 struct pal_in4_addr peerId, int VirtualLinkInstanceId)`

This function configures the instance ID on the specified virtual link. `smi_ospf6_vlink_instance_id_set`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *areaId* OSPFv3 area ID in an IPv4 address format
 ← *peerId* Neighbor ID
 ← *instanceId* Instance ID numeric <0-255>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.1.2.290 `int smi_ospf6_vlink_instance_id_unset (struct smiclient_globals * azg,
 u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId,
 struct pal_in4_addr peerId)`

This function resets the configured instance ID of the specified virtual link to the default value as 0. `smi_ospf6_vlink_instance_id_unset`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *areaId* OSPFv3 area ID in an IPv4 address format
 ← *peerId* Neighbor ID

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.1.2.291 `int smi_ospf6_vlink_retransmit_interval_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkRetransmitInterval)`

This function sets the retransmit interval on the specified virtual link. `smi_ospf6_vlink_retransmit_interval_set`

Parameters:

← **azg** Pointer to the SMI client global structure
 ← **vrId** Virtual Router ID numeric <0-255>
 ← **ospf6ProcessTag** OSPFv3 Process tag
 ← **areaId** OSPFv3 area ID in an IPv4 address format
 ← **peerId** Neighbor ID
 ← **virtualLinkRetransmitInterval** Retransmit Interval in seconds <1-65535>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_IF_RETRANSMIT_INTERVAL_INVALID
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.1.2.292 `int smi_ospf6_vlink_retransmit_interval_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId)`

This This function resets the configured retransmit interval of the specified virtual link to the default value as 5. `smi_ospf6_vlink_retransmit_interval_unset`

Parameters:

← **azg** Pointer to the SMI client global structure
 ← **vrId** Virtual Router ID numeric <0-255>
 ← **ospf6ProcessTag** OSPFv3 Process tag
 ← **areaId** OSPFv3 area ID in an IPv4 address format
 ← **peerId** Neighbor ID

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.1.2.293 `int smi_ospf6_vlink_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int ospf6AreaFormat)`

This function configures the virtual link as specified. `smi_ospf6_vlink_set`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *areaId* OSPFv3 area ID in an IPv4 address format
 ← *peerId* Neighbor ID.

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_AREA_NOT_DEFAULT
 OSPF6_API_SET_ERR_VLINK_CANT_GET

2.1.2.294 `int smi_ospf6_vlink_transmit_delay_set (struct smiclient_globals * azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtualLinkTransmitDelay)`

This function configures the transmit delay on a specified virtual link. `smi_ospf6_vlink_transmit_delay_set`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *areaId* OSPFv3 area ID in an IPv4 address format
 ← *peerId* Neighbor ID
 ← *vTransmitDelay* Transmission delay in seconds <1-65535>

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_IF_TRANSMIT_DELAY_INVALID
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.1.2.295 `int smi_ospf6_vlink_transmit_delay_unset (struct smiclient_globals *
azg, u_int32_t vrId, char * ospf6ProcessTag, struct pal_in4_addr
areaId, struct pal_in4_addr peerId)`

This function unsets the configured transmit delay on a specified virtual link. `smi_ospf6_vlink_transmit_delay_unset`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *areaId* OSPFv3 area ID in an IPv4 address format
 ← *peerId* Neighbor ID

Returns:

0 on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.1.2.296 `int smi_ospf6_vlink_unset (struct smiclient_globals * azg, u_int32_t
vrId, char * ospf6ProcessTag, struct pal_in4_addr areaId, struct
 pal_in4_addr peerId)`

This function removes configuration of the virtual link as specified. `smi_ospf6_vlink_unset`

Parameters:

← *azg* Pointer to the SMI client global structure
 ← *vrId* Virtual Router ID numeric <0-255>
 ← *ospf6ProcessTag* OSPFv3 Process tag
 ← *areaId* OSPFv3 area ID in an IPv4 address format
 ← *peerId* Neighbor ID.

Returns:

0 on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.2 smi_ospf6_bfd.h File Reference

Provides APIs for managing Bidirectional Forwarding Detection(BFD) in ZebOS.

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

```
#include "smi_ospf6_bfd_msg.h"
```

Functions

- int [smi_ospf6_if_bfd_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, int instance_id)

This function Sets the BFD fall-over check for neighbors on specified interface.

- int [smi_ospf6_if_bfd_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, int instance_id)

This function Unsets the BFD fall-over check for neighbors on specified interface.

- int [smi_ospf6_if_bfd_disable_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, int instance_id)

This function Disables the BFD fall-over check for neighbors on specified interface.

- int [smi_ospf6_if_bfd_disable_set_validate](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, int instance_id)

- int [smi_ospf6_if_bfd_disable_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, int instance_id)

This function Unsets the disable flag of BFD fall-over check for neighbors on specified interface.

- int [smi_ospf6_if_bfd_disable_unset_validate](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *ifname, int instance_id)

- int [smi_ospf6_bfd_all_interfaces_set_validate](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function Sets the BFD fall-over check for all the neighbors under specified process.

- int [smi_ospf6_bfd_all_interfaces_set](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

- int [smi_ospf6_bfd_all_interfaces_set_wrap_validate](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int8_t bfdAllInterface)

- int [smi_ospf6_bfd_all_interfaces_set_wrap](#) (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int8_t bfdAllInterface)

- int [smi_ospf6_bfd_all_interfaces_unset_validate](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function Unsets the BFD fall-over check for all the neighbors under specified process.

- int [smi_ospf6_bfd_all_interfaces_unset](#) (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

- int **smi_ospf6_bfd_all_interfaces_unset_wrap_validate** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int8_t bfdAllInterface)
- int **smi_ospf6_bfd_all_interfaces_unset_wrap** (struct smiclient_globals *azg, u_int32_t vrId, char *ospf6ProcessTag, u_int8_t bfdAllInterface)
- int **smi_ospf6_vlink_bfd_set** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, struct pal_in4_addr area_id, struct pal_in4_addr peer_id)

This function Sets the BFD fall-over check for the specified VLINK neighbor.

- int **smi_ospf6_vlink_bfd_unset** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag, struct pal_in4_addr area_id, struct pal_in4_addr peer_id)

This function Unset the BFD fall-over check for the specified VLINK neighbor.

- int **smi_ospf6_vlink_bfd_all_set_sdkapi** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function.

- int **smi_ospf6_vlink_bfd_all_unset_sdkapi** (struct smiclient_globals *azg, u_int32_t vr_id, char *tag)

This function.

2.2.1 Detailed Description

Provides APIs for managing Bidirectional Forwarding Detection(BFD) in ZebOS.

2.2.2 Function Documentation

2.2.2.1 int smi_ospf6_bfd_all_interfaces_set_validate (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)

This function Sets the BFD fall-over check for all the neighbors under specified process. smi_ospf6_bfd_all_interfaces_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* OSPFv3 process tag

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.2.2.2 `int smi_ospf6_bfd_all_interfaces_unset_validate (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function Unsets the BFD fall-over check for all the neighbors under specified process. `smi_ospf6_bfd_all_interfaces_unset`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* OSPFv3 process tag

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST

2.2.2.3 `int smi_ospf6_if_bfd_disable_set (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname, int instance_id)`

This function Disables the BFD fall-over check for neighbors on specified interface. `smi_ospf6_if_bfd_disable_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *ifname* The Interface name
- ← *instance_id* Interface instance ID <0-255>.

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID

2.2.2.4 `int smi_ospf6_if_bfd_disable_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname, int instance_id)`

This function Unsets the disable flag of BFD fall-over check for neighbors on specified interface. `smi_ospf6_if_bfd_disable_unset`

Parameters:

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

- ← *ifname* The Interface name
- ← *instance_id* Interface instance ID <0-255>.

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID

2.2.2.5 int smi_ospf6_if_bfd_set (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname, int instance_id)

This function Sets the BFD fall-over check for neighbors on specifcied interface. smi_ospf6_if_bfd_set

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *ifname* The Interface name
- ← *instance_id* Interface instance ID <0-255>.

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID

2.2.2.6 int smi_ospf6_if_bfd_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname, int instance_id)

This function Unsets the BFD fall-over check for neighbors on specifcied interface. smi_ospf6_if_bfd_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *ifname* The Interface name
- ← *instance_id* Interface instance ID <0-255>.

Returns:

OSPF6_API_SET_SUCCESS on success, otherwise one of the following error codes
OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID

2.2.2.7 `int smi_ospf6_vlink_bfd_all_set_sdkapi (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function. `smi_ospf6_vlink_bfd_all_set_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router ID numeric <0-255>
- ← *tag* OSPFv3 process tag

Returns:

OSPF6_API_SET_SUCCESS on success

2.2.2.8 `int smi_ospf6_vlink_bfd_all_unset_sdkapi (struct smiclient_globals * azg, u_int32_t vr_id, char * tag)`

This function. `smi_ospf6_vlink_bfd_all_unset_sdkapi`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual router ID; for a non-virtual-router implementation, specify 0
- ← *tag* OSPFv3 process tag

Returns:

OSPF6_API_SET_SUCCESS on success

2.2.2.9 `int smi_ospf6_vlink_bfd_set (struct smiclient_globals * azg, u_int32_t vr_id, char * tag, struct pal_in4_addr area_id, struct pal_in4_addr peer_id)`

This function Sets the BFD fall-over check for the specified VLINK neighbor. `smi_ospf6_vlink_bfd_set`

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *area_id* OSPFv3 area ID in an IPv4 address format
- ← *peer_id* Neighbor ID

Returns:

vr_id, *vlink->name*, *vlink->instance_id* on success, otherwise one of the following error codes
 OSPF6_API_SET_ERR_VR_NOT_EXIST
 OSPF6_API_SET_ERR_PROCESS_NOT_EXIST
 OSPF6_API_SET_ERR_VLINK_NOT_EXIST

2.2.2.10 `int smi_ospf6_vlink_bfd_unset (struct smiclient_globals * azg,
u_int32_t vr_id, char * tag, struct pal_in4_addr area_id, struct
pal_in4_addr peer_id)`

This function Unset the BFD fall-over check for the specified VLINK neighbor. smi_ospf6_vlink_bfd_unset

Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr_id* Virtual Router Id
- ← *tag* OSPFv3 Process tag
- ← *area_id* OSPFv3 area ID in an IPv4 address format
- ← *peer_id* Neighbor ID

Returns:

vr_id, vlink->name, vlink->instance_id on success, otherwise one of the following error codes OSPF6_API_SET_ERR_VR_NOT_EXIST
OSPF6_API_SET_ERR_IF_INSTANCE_ID_INVALID
OSPF6_API_SET_ERR_VLINK_NOT_EXIST

Index

- merge_ospf6_list_nbr_info
 - smi_ospf6.h, [46](#)
- smi_debug_no_ospf6_packet
 - smi_ospf6.h, [46](#)
- smi_debug_ospf6_packet
 - smi_ospf6.h, [47](#)
- smi_ospf6.h, [3](#)
 - merge_ospf6_list_nbr_info, [46](#)
 - smi_debug_no_ospf6_packet, [46](#)
 - smi_debug_ospf6_packet, [47](#)
 - smi_ospf6_abr_type_set, [48](#)
 - smi_ospf6_abr_type_unset, [48](#)
 - smi_ospf6_address_family_set, [49](#)
 - smi_ospf6_address_family_unset, [49](#)
 - smi_ospf6_area_default_cost_set, [49](#)
 - smi_ospf6_area_default_cost_unset, [50](#)
 - smi_ospf6_area_format_set, [50](#)
 - smi_ospf6_area_no_summary_set, [51](#)
 - smi_ospf6_area_nssa_default_ -
originate_metric_set, [51](#)
 - smi_ospf6_area_nssa_default_ -
originate_metric_type_set, [52](#)
 - smi_ospf6_area_nssa_default_ -
originate_set, [52](#)
 - smi_ospf6_area_nssa_default_ -
originate_unset, [53](#)
 - smi_ospf6_area_nssa_no_ -
redistribution_set, [53](#)
 - smi_ospf6_area_nssa_no_ -
redistribution_unset, [54](#)
 - smi_ospf6_area_nssa_set, [54](#)
 - smi_ospf6_area_nssa_stability_ -
interval_set, [55](#)
 - smi_ospf6_area_nssa_translator_ -
role_set, [55](#)
 - smi_ospf6_area_nssa_translator_ -
role_unset, [56](#)
 - smi_ospf6_area_nssa_unset, [56](#)
 - smi_ospf6_area_range_ipv4_set_ -
sdkapi, [57](#)
 - smi_ospf6_area_range_ipv4_ -
unset_sdkapi, [58](#)
 - smi_ospf6_area_stub_no_ -
summary_set, [59](#)
 - smi_ospf6_area_stub_set, [59](#)
 - smi_ospf6_area_stub_unset, [60](#)
 - smi_ospf6_auto_cost_reference_ -
bandwidth_set, [60](#)
 - smi_ospf6_auto_cost_reference_ -
bandwidth_unset, [61](#)
 - smi_ospf6_capability_cspf_set, [61](#)
 - smi_ospf6_capability_cspf_unset, [62](#)
 - smi_ospf6_capability_restart_set, [62](#)
 - smi_ospf6_capability_restart_unset, [62](#)
 - smi_ospf6_debug, [63](#)
 - smi_ospf6_default_metric_set, [64](#)
 - smi_ospf6_default_metric_unset, [65](#)
 - smi_ospf6_disable_db_summary_ -
opt, [65](#)
 - smi_ospf6_distance_all_set, [65](#)
 - smi_ospf6_distance_all_unset, [66](#)
 - smi_ospf6_distance_external_set, [66](#)
 - smi_ospf6_distance_external_unset, [67](#)
 - smi_ospf6_distance_inter_area_set, [67](#)
 - smi_ospf6_distance_inter_area_ -
unset, [67](#)
 - smi_ospf6_distance_intra_area_set, [68](#)
 - smi_ospf6_distance_intra_area_ -
unset, [68](#)
 - smi_ospf6_distribute_list_in_set, [69](#)

- smi_ospf6_distribute_list_in_unset, 69
- smi_ospf6_distribute_list_out_set, 69
- smi_ospf6_distribute_list_out_unset, 70
- smi_ospf6_enable_db_summary_opt, 71
- smi_ospf6_get_admin_stat, 71
- smi_ospf6_get_area_bdr_rtr_count, 72
- smi_ospf6_get_area_bdr_rtr_status, 72
- smi_ospf6_get_area_lsa_cksum_sum, 72
- smi_ospf6_get_area_lsa_count, 73
- smi_ospf6_get_area_lsdbs_advertisement, 73
- smi_ospf6_get_area_lsdbs_age, 74
- smi_ospf6_get_area_lsdbs_checksum, 74
- smi_ospf6_get_area_lsdbs_sequence, 75
- smi_ospf6_get_area_lsdbs_type_known, 75
- smi_ospf6_get_area_nssa_trans_events, 76
- smi_ospf6_get_area_nssa_trans_role, 76
- smi_ospf6_get_area_nssa_trans_stability_interval, 77
- smi_ospf6_get_area_nssa_trans_state, 77
- smi_ospf6_get_area_status, 78
- smi_ospf6_get_area_stub_metric_type, 78
- smi_ospf6_get_area_summary, 78
- smi_ospf6_get_area_te_enabled, 79
- smi_ospf6_get_as_lsdbs_advertisement, 79
- smi_ospf6_get_as_lsdbs_age, 80
- smi_ospf6_get_as_lsdbs_checksum, 80
- smi_ospf6_get_as_lsdbs_sequence, 81
- smi_ospf6_get_as_lsdbs_type_known, 81
- smi_ospf6_get_as_scope_lsa_cksumsum, 81
- smi_ospf6_get_as_scope_lsa_count, 82
- smi_ospf6_get_asbdr_rtr_count, 82
- smi_ospf6_get_asbdr_rtr_status, 83
- smi_ospf6_get_discontinuity_time, 83
- smi_ospf6_get_extern_lsa_count, 83
- smi_ospf6_get_if_admin_stat, 84
- smi_ospf6_get_if_area_id, 84
- smi_ospf6_get_if_bdr, 85
- smi_ospf6_get_if_demand, 85
- smi_ospf6_get_if_demand_nbr_probe, 85
- smi_ospf6_get_if_demand_nbr_probe_interval, 86
- smi_ospf6_get_if_demand_nbr_probe_retrans_limit, 86
- smi_ospf6_get_if_dr, 87
- smi_ospf6_get_if_events, 87
- smi_ospf6_get_if_hello_interval, 88
- smi_ospf6_get_if_link_lsa_cksumsum, 88
- smi_ospf6_get_if_link_lsa_suppression, 88
- smi_ospf6_get_if_link_scope_lsa_count, 89
- smi_ospf6_get_if_metric_value, 89
- smi_ospf6_get_if_poll_interval, 90
- smi_ospf6_get_if_retrans_interval, 90
- smi_ospf6_get_if_rtr_dead_interval, 91
- smi_ospf6_get_if_rtr_priority, 91
- smi_ospf6_get_if_state, 91
- smi_ospf6_get_if_status, 92
- smi_ospf6_get_if_te_disabled, 92
- smi_ospf6_get_if_transit_delay, 93
- smi_ospf6_get_if_type, 93
- smi_ospf6_get_import_as_extern, 94
- smi_ospf6_get_link_lsdbs_advertisement, 94
- smi_ospf6_get_link_lsdbs_age, 95
- smi_ospf6_get_link_lsdbs_checksum, 95
- smi_ospf6_get_link_lsdbs_sequence, 96
- smi_ospf6_get_link_lsdbs_type_known, 96
- smi_ospf6_get_nbr_address, 97

- smi_ospf6_get_nbr_address_type, 97
- smi_ospf6_get_nbr_events, 98
- smi_ospf6_get_nbr_hello_-suppressed, 98
- smi_ospf6_get_nbr_if_id, 99
- smi_ospf6_get_nbr_lsretransq_len, 99
- smi_ospf6_get_nbr_options, 100
- smi_ospf6_get_nbr_priority, 100
- smi_ospf6_get_nbr_restart_helper_-age, 100
- smi_ospf6_get_nbr_restart_helper_-exit_reason, 101
- smi_ospf6_get_nbr_restart_helper_-status, 101
- smi_ospf6_get_nbr_state, 102
- smi_ospf6_get_notification_enable, 102
- smi_ospf6_get_originate_new_lsas, 103
- smi_ospf6_get_reference_-bandwidth, 103
- smi_ospf6_get_restart_age, 103
- smi_ospf6_get_restart_exit_reason, 104
- smi_ospf6_get_restart_interval, 104
- smi_ospf6_get_restart_status, 105
- smi_ospf6_get_restart_strict_lsa_-check, 105
- smi_ospf6_get_restart_support, 105
- smi_ospf6_get_restart_time, 106
- smi_ospf6_get_router_id, 106
- smi_ospf6_get_rx_new_lsas, 106
- smi_ospf6_get_spf_runs, 107
- smi_ospf6_get_stub_metric, 107
- smi_ospf6_get_stub_router_-advertisement, 108
- smi_ospf6_get_stub_router_support, 108
- smi_ospf6_get_version_num, 108
- smi_ospf6_get_virt_if_events, 109
- smi_ospf6_get_virt_if_hello_-interval, 109
- smi_ospf6_get_virt_if_index, 109
- smi_ospf6_get_virt_if_instid, 110
- smi_ospf6_get_virt_if_link_lsa_-cksumsum, 110
- smi_ospf6_get_virt_if_link_scope_-lsa_count, 111
- smi_ospf6_get_virt_if_retrans_-interval, 111
- smi_ospf6_get_virt_if_rtr_dead_-interval, 112
- smi_ospf6_get_virt_if_state, 112
- smi_ospf6_get_virt_if_status, 113
- smi_ospf6_get_virt_if_transmit_-delay, 113
- smi_ospf6_get_virt_link_lsdb_-advertisement, 113
- smi_ospf6_get_virt_link_lsdb_age, 114
- smi_ospf6_get_virt_link_lsdb_-checksum, 114
- smi_ospf6_get_virt_link_lsdb_-sequence, 115
- smi_ospf6_get_virt_link_lsdb_-type_known, 116
- smi_ospf6_get_virt_nbr_address, 116
- smi_ospf6_get_virt_nbr_address_-type, 117
- smi_ospf6_get_virt_nbr_events, 117
- smi_ospf6_get_virt_nbr_hello_-suppressed, 118
- smi_ospf6_get_virt_nbr_if_id, 118
- smi_ospf6_get_virt_nbr_ifindex, 119
- smi_ospf6_get_virt_nbr_ifinstid, 119
- smi_ospf6_get_virt_nbr_ls_-retransq_len, 120
- smi_ospf6_get_virt_nbr_options, 120
- smi_ospf6_get_virt_nbr_restart_-helper_age, 121
- smi_ospf6_get_virt_nbr_restart_-helper_exit_reason, 121
- smi_ospf6_get_virt_nbr_restart_-helper_status, 122
- smi_ospf6_get_virt_nbr_state, 122
- smi_ospf6_graceful_restart_set, 123
- smi_ospf6_graceful_restart_unset, 123
- smi_ospf6_if_cost_set, 123
- smi_ospf6_if_cost_unset, 124
- smi_ospf6_if_dead_interval_set, 124
- smi_ospf6_if_dead_interval_unset, 125

- smi_ospf6_if_disable_all_set, [125](#)
- smi_ospf6_if_disable_all_unset, [125](#)
- smi_ospf6_if_hello_interval_set, [126](#)
- smi_ospf6_if_hello_interval_unset, [126](#)
- smi_ospf6_if_ipv6_router_set, [127](#)
- smi_ospf6_if_ipv6_router_unset, [127](#)
- smi_ospf6_if_link_lsa_-
suppression_set, [128](#)
- smi_ospf6_if_mtu_ignore_set, [129](#)
- smi_ospf6_if_mtu_ignore_unset, [129](#)
- smi_ospf6_if_network_set, [130](#)
- smi_ospf6_if_network_unset, [130](#)
- smi_ospf6_if_priority_set, [130](#)
- smi_ospf6_if_priority_unset, [131](#)
- smi_ospf6_if_retransmit_interval_-
set, [131](#)
- smi_ospf6_if_retransmit_interval_-
unset, [132](#)
- smi_ospf6_if_te_metric_set, [132](#)
- smi_ospf6_if_te_metric_unset, [133](#)
- smi_ospf6_if_transmit_delay_set, [133](#)
- smi_ospf6_if_transmit_delay_unset, [134](#)
- smi_ospf6_ipv6_ospf_display_-
route_single_line_set, [134](#)
- smi_ospf6_ipv6_ospf_display_-
route_single_line_unset, [134](#)
- smi_ospf6_log_adjacency_-
changes_set, [135](#)
- smi_ospf6_log_adjacency_-
changes_unset, [135](#)
- smi_ospf6_max_concurrent_dd_set, [136](#)
- smi_ospf6_max_concurrent_dd_-
unset, [136](#)
- smi_ospf6_max_unuse_lsa_set, [136](#)
- smi_ospf6_max_unuse_lsa_unset, [137](#)
- smi_ospf6_max_unuse_packet_set, [137](#)
- smi_ospf6_max_unuse_packet_-
unset, [138](#)
- smi_ospf6_no_debug, [138](#)
- smi_ospf6_passive_if_default_set, [140](#)
- smi_ospf6_passive_if_default_-
unset, [140](#)
- smi_ospf6_passive_if_set, [140](#)
- smi_ospf6_passive_if_unset, [141](#)
- smi_ospf6_process_shutdown_set, [141](#)
- smi_ospf6_process_shutdown_-
unset, [141](#)
- smi_ospf6_redist_default_set, [142](#)
- smi_ospf6_redist_default_unset, [142](#)
- smi_ospf6_redistribute_metric_-
set_by_ospf6_src_tag_sdkapi, [143](#)
- smi_ospf6_redistribute_metric_-
set_by_ospf_src_tag_sdkapi, [143](#)
- smi_ospf6_redistribute_metric_-
type_set_by_ospf6_src_tag_-
sdkapi, [144](#)
- smi_ospf6_redistribute_metric_-
type_set_by_ospf_src_tag_-
sdkapi, [145](#)
- smi_ospf6_redistribute_metric_-
type_unset_by_ospf6_src_-
tag_sdkapi, [145](#)
- smi_ospf6_redistribute_metric_-
type_unset_by_ospf_src_tag_-
sdkapi, [146](#)
- smi_ospf6_redistribute_metric_-
unset_by_ospf6_src_tag_-
sdkapi, [147](#)
- smi_ospf6_redistribute_metric_-
unset_by_ospf_src_tag_sdkapi, [147](#)
- smi_ospf6_redistribute_route_tag_-
set_by_ospf6_src_tag_sdkapi, [148](#)
- smi_ospf6_redistribute_route_tag_-
set_by_ospf_src_tag_sdkapi, [149](#)
- smi_ospf6_redistribute_route_-
tag_unset_by_ospf6_src_tag_-
sdkapi, [149](#)
- smi_ospf6_redistribute_route_tag_-
unset_by_ospf_src_tag_sdkapi, [150](#)
- smi_ospf6_redistribute_set_by_-
ospf6_src_tag_sdkapi, [151](#)
- smi_ospf6_redistribute_set_by_-
ospf_src_tag_sdkapi, [151](#)

- smi_ospf6_redistribute_unset_-
by_ospf6_src_tag_sdkapi,
152
- smi_ospf6_redistribute_unset_by_-
ospf_src_tag_sdkapi, 152
- smi_ospf6_restart_graceful_set, 153
- smi_ospf6_restart_helper_grace_-
period_set, 154
- smi_ospf6_restart_helper_grace_-
period_unset, 154
- smi_ospf6_restart_helper_never_-
router_id_set, 154
- smi_ospf6_restart_helper_never_-
router_id_unset, 155
- smi_ospf6_restart_helper_never_-
router_unset_all, 155
- smi_ospf6_restart_helper_policy_-
set, 155
- smi_ospf6_restart_helper_policy_-
unset, 156
- smi_ospf6_restart_helper_policy_-
unset_all, 156
- smi_ospf6_routemap_default_set,
157
- smi_ospf6_routemap_default_unset,
157
- smi_ospf6_routemap_set_by_-
ospf6_src_tag_sdkapi, 157
- smi_ospf6_routemap_set_by_ospf_-
src_tag_sdkapi, 158
- smi_ospf6_routemap_unset_by_-
ospf6_src_tag_sdkapi, 159
- smi_ospf6_routemap_unset_by_-
ospf_src_tag_sdkapi, 159
- smi_ospf6_router_id_set, 160
- smi_ospf6_router_id_unset, 160
- smi_ospf6_router_set, 161
- smi_ospf6_router_unset, 161
- smi_ospf6_set_area_nssa_trans_-
role, 161
- smi_ospf6_set_area_stub_metric_-
type, 162
- smi_ospf6_set_area_te_enabled, 163
- smi_ospf6_set_asbdr_rtr_status, 163
- smi_ospf6_set_if_admin_stat, 163
- smi_ospf6_set_if_rtr_dead_interval,
164
- smi_ospf6_set_if_type, 165
- smi_ospf6_show_border_routers,
165
- smi_ospf6_show_database, 166
- smi_ospf6_show_debug, 166
- smi_ospf6_show_interface, 166
- smi_ospf6_show_interface_brief,
167
- smi_ospf6_show_interface_brief_-
by_tag, 167
- smi_ospf6_show_interface_by_tag,
168
- smi_ospf6_show_neighbor_by_if_-
nbr_id, 168
- smi_ospf6_show_neighbor_by_-
interface_name, 169
- smi_ospf6_show_neighbor_by_-
nbr_id, 169
- smi_ospf6_show_neighbor_detail_-
by_if_nbr_id, 170
- smi_ospf6_show_neighbor_detail_-
by_interface_name, 170
- smi_ospf6_show_neighbor_detail_-
by_tag, 171
- smi_ospf6_show_neighbor_-
summary_by_if_nbr_id, 171
- smi_ospf6_show_neighbor_-
summary_by_interface_name,
172
- smi_ospf6_show_neighbor_-
summary_by_nbr_id, 172
- smi_ospf6_show_neighbor_-
summary_by_tag, 173
- smi_ospf6_show_process, 173
- smi_ospf6_show_route, 174
- smi_ospf6_show_route_by_prefix,
174
- smi_ospf6_show_route_summary,
175
- smi_ospf6_show_virtual_link, 175
- smi_ospf6_show_virtual_link_brief,
176
- smi_ospf6_summary_address_-
ipv4_set_sdkapi, 176
- smi_ospf6_summary_address_-
ipv4_unset_sdkapi, 177
- smi_ospf6_summary_address_not_-
advertise_ipv4_set_sdkapi,
178
- smi_ospf6_summary_address_not_-
advertise_ipv4_unset_sdkapi,
179

- smi_ospf6_summary_address_not_advertise_ipv6_unset_sdkapi, [179](#)
- smi_ospf6_summary_address_tag_ipv4_set_sdkapi, [180](#)
- smi_ospf6_summary_address_tag_ipv4_unset_sdkapi, [181](#)
- smi_ospf6_timers_spf_set, [182](#)
- smi_ospf6_timers_spf_unset, [182](#)
- smi_ospf6_vlink_dead_interval_set, [183](#)
- smi_ospf6_vlink_dead_interval_unset, [183](#)
- smi_ospf6_vlink_hello_interval_set, [184](#)
- smi_ospf6_vlink_hello_interval_unset, [184](#)
- smi_ospf6_vlink_instance_id_set, [185](#)
- smi_ospf6_vlink_instance_id_unset, [185](#)
- smi_ospf6_vlink_retransmit_interval_set, [186](#)
- smi_ospf6_vlink_retransmit_interval_unset, [186](#)
- smi_ospf6_vlink_set, [187](#)
- smi_ospf6_vlink_transmit_delay_set, [187](#)
- smi_ospf6_vlink_transmit_delay_unset, [188](#)
- smi_ospf6_vlink_unset, [188](#)
- smi_ospf6_abr_type_set
smi_ospf6.h, [48](#)
- smi_ospf6_abr_type_unset
smi_ospf6.h, [48](#)
- smi_ospf6_address_family_set
smi_ospf6.h, [49](#)
- smi_ospf6_address_family_unset
smi_ospf6.h, [49](#)
- smi_ospf6_area_default_cost_set
smi_ospf6.h, [49](#)
- smi_ospf6_area_default_cost_unset
smi_ospf6.h, [50](#)
- smi_ospf6_area_format_set
smi_ospf6.h, [50](#)
- smi_ospf6_area_no_summary_set
smi_ospf6.h, [51](#)
- smi_ospf6_area_nssa_default_originate_metric_set
smi_ospf6.h, [51](#)
- smi_ospf6_area_nssa_default_originate_metric_type_set
smi_ospf6.h, [52](#)
- smi_ospf6_area_nssa_default_originate_set
smi_ospf6.h, [52](#)
- smi_ospf6_area_nssa_default_originate_unset
smi_ospf6.h, [53](#)
- smi_ospf6_area_nssa_no_redistribution_set
smi_ospf6.h, [53](#)
- smi_ospf6_area_nssa_no_redistribution_unset
smi_ospf6.h, [54](#)
- smi_ospf6_area_nssa_set
smi_ospf6.h, [54](#)
- smi_ospf6_area_nssa_stability_interval_set
smi_ospf6.h, [55](#)
- smi_ospf6_area_nssa_translator_role_set
smi_ospf6.h, [55](#)
- smi_ospf6_area_nssa_translator_role_unset
smi_ospf6.h, [56](#)
- smi_ospf6_area_nssa_unset
smi_ospf6.h, [56](#)
- smi_ospf6_area_range_ipv4_set_sdkapi
smi_ospf6.h, [57](#)
- smi_ospf6_area_range_ipv4_unset_sdkapi
smi_ospf6.h, [58](#)
- smi_ospf6_area_stub_no_summary_set
smi_ospf6.h, [59](#)
- smi_ospf6_area_stub_set
smi_ospf6.h, [59](#)
- smi_ospf6_area_stub_unset
smi_ospf6.h, [60](#)
- smi_ospf6_auto_cost_reference_bandwidth_set
smi_ospf6.h, [60](#)
- smi_ospf6_auto_cost_reference_bandwidth_unset
smi_ospf6.h, [61](#)
- smi_ospf6_bfd.h, [190](#)
- smi_ospf6_bfd_all_interfaces_set_validate, [191](#)
- smi_ospf6_bfd_all_interfaces_unset_validate, [191](#)
- smi_ospf6_if_bfd_disable_set, [192](#)

- smi_ospf6_if_bfd_disable_unset, 192
- smi_ospf6_if_bfd_set, 193
- smi_ospf6_if_bfd_unset, 193
- smi_ospf6_vlink_bfd_all_set - sdkapi, 193
- smi_ospf6_vlink_bfd_all_unset - sdkapi, 194
- smi_ospf6_vlink_bfd_set, 194
- smi_ospf6_vlink_bfd_unset, 194
- smi_ospf6_bfd_all_interfaces_set - validate
smi_ospf6_bfd.h, 191
- smi_ospf6_bfd_all_interfaces_unset - validate
smi_ospf6_bfd.h, 191
- smi_ospf6_capability_cspf_set
smi_ospf6.h, 61
- smi_ospf6_capability_cspf_unset
smi_ospf6.h, 62
- smi_ospf6_capability_restart_set
smi_ospf6.h, 62
- smi_ospf6_capability_restart_unset
smi_ospf6.h, 62
- smi_ospf6_debug
smi_ospf6.h, 63
- smi_ospf6_default_metric_set
smi_ospf6.h, 64
- smi_ospf6_default_metric_unset
smi_ospf6.h, 65
- smi_ospf6_disable_db_summary_opt
smi_ospf6.h, 65
- smi_ospf6_distance_all_set
smi_ospf6.h, 65
- smi_ospf6_distance_all_unset
smi_ospf6.h, 66
- smi_ospf6_distance_external_set
smi_ospf6.h, 66
- smi_ospf6_distance_external_unset
smi_ospf6.h, 67
- smi_ospf6_distance_inter_area_set
smi_ospf6.h, 67
- smi_ospf6_distance_inter_area_unset
smi_ospf6.h, 67
- smi_ospf6_distance_intra_area_set
smi_ospf6.h, 68
- smi_ospf6_distance_intra_area_unset
smi_ospf6.h, 68
- smi_ospf6_distribute_list_in_set
smi_ospf6.h, 69
- smi_ospf6_distribute_list_in_unset
smi_ospf6.h, 69
- smi_ospf6_distribute_list_out_set
smi_ospf6.h, 69
- smi_ospf6_distribute_list_out_unset
smi_ospf6.h, 70
- smi_ospf6_enable_db_summary_opt
smi_ospf6.h, 71
- smi_ospf6_get_admin_stat
smi_ospf6.h, 71
- smi_ospf6_get_area_bdr_rtr_count
smi_ospf6.h, 72
- smi_ospf6_get_area_bdr_rtr_status
smi_ospf6.h, 72
- smi_ospf6_get_area_lsa_cksum_sum
smi_ospf6.h, 72
- smi_ospf6_get_area_lsa_count
smi_ospf6.h, 73
- smi_ospf6_get_area_lsdb_advertisement
smi_ospf6.h, 73
- smi_ospf6_get_area_lsdb_age
smi_ospf6.h, 74
- smi_ospf6_get_area_lsdb_checksum
smi_ospf6.h, 74
- smi_ospf6_get_area_lsdb_sequence
smi_ospf6.h, 75
- smi_ospf6_get_area_lsdb_type_known
smi_ospf6.h, 75
- smi_ospf6_get_area_nssa_trans_events
smi_ospf6.h, 76
- smi_ospf6_get_area_nssa_trans_role
smi_ospf6.h, 76
- smi_ospf6_get_area_nssa_trans - stability_interval
smi_ospf6.h, 77
- smi_ospf6_get_area_nssa_trans_state
smi_ospf6.h, 77
- smi_ospf6_get_area_status
smi_ospf6.h, 78
- smi_ospf6_get_area_stub_metric_type
smi_ospf6.h, 78
- smi_ospf6_get_area_summary
smi_ospf6.h, 78
- smi_ospf6_get_area_te_enabled
smi_ospf6.h, 79
- smi_ospf6_get_as_lsdb_advertisement
smi_ospf6.h, 79
- smi_ospf6_get_as_lsdb_age
smi_ospf6.h, 80
- smi_ospf6_get_as_lsdb_checksum

- smi_ospf6.h, [80](#)
- smi_ospf6_get_as_lsdb_sequence
 - smi_ospf6.h, [81](#)
- smi_ospf6_get_as_lsdb_type_known
 - smi_ospf6.h, [81](#)
- smi_ospf6_get_as_scope_lsa_cksumsum
 - smi_ospf6.h, [81](#)
- smi_ospf6_get_as_scope_lsa_count
 - smi_ospf6.h, [82](#)
- smi_ospf6_get_asbdr_rtr_count
 - smi_ospf6.h, [82](#)
- smi_ospf6_get_asbdr_rtr_status
 - smi_ospf6.h, [83](#)
- smi_ospf6_get_discontinuity_time
 - smi_ospf6.h, [83](#)
- smi_ospf6_get_extern_lsa_count
 - smi_ospf6.h, [83](#)
- smi_ospf6_get_if_admin_stat
 - smi_ospf6.h, [84](#)
- smi_ospf6_get_if_area_id
 - smi_ospf6.h, [84](#)
- smi_ospf6_get_if_bdr
 - smi_ospf6.h, [85](#)
- smi_ospf6_get_if_demand
 - smi_ospf6.h, [85](#)
- smi_ospf6_get_if_demand_nbr_probe
 - smi_ospf6.h, [85](#)
- smi_ospf6_get_if_demand_nbr_probe - interval
 - smi_ospf6.h, [86](#)
- smi_ospf6_get_if_demand_nbr_probe - retrans_limit
 - smi_ospf6.h, [86](#)
- smi_ospf6_get_if_dr
 - smi_ospf6.h, [87](#)
- smi_ospf6_get_if_events
 - smi_ospf6.h, [87](#)
- smi_ospf6_get_if_hello_interval
 - smi_ospf6.h, [88](#)
- smi_ospf6_get_if_link_lsa_cksumsum
 - smi_ospf6.h, [88](#)
- smi_ospf6_get_if_link_lsa_suppression
 - smi_ospf6.h, [88](#)
- smi_ospf6_get_if_link_scope_lsa_count
 - smi_ospf6.h, [89](#)
- smi_ospf6_get_if_metric_value
 - smi_ospf6.h, [89](#)
- smi_ospf6_get_if_poll_interval
 - smi_ospf6.h, [90](#)
- smi_ospf6_get_if_retrans_interval
 - smi_ospf6.h, [90](#)
- smi_ospf6_get_if_rtr_dead_interval
 - smi_ospf6.h, [91](#)
- smi_ospf6_get_if_rtr_priority
 - smi_ospf6.h, [91](#)
- smi_ospf6_get_if_state
 - smi_ospf6.h, [91](#)
- smi_ospf6_get_if_status
 - smi_ospf6.h, [92](#)
- smi_ospf6_get_if_te_disabled
 - smi_ospf6.h, [92](#)
- smi_ospf6_get_if_transit_delay
 - smi_ospf6.h, [93](#)
- smi_ospf6_get_if_type
 - smi_ospf6.h, [93](#)
- smi_ospf6_get_import_as_extern
 - smi_ospf6.h, [94](#)
- smi_ospf6_get_link_lsdb_advertisement
 - smi_ospf6.h, [94](#)
- smi_ospf6_get_link_lsdb_age
 - smi_ospf6.h, [95](#)
- smi_ospf6_get_link_lsdb_checksum
 - smi_ospf6.h, [95](#)
- smi_ospf6_get_link_lsdb_sequence
 - smi_ospf6.h, [96](#)
- smi_ospf6_get_link_lsdb_type_known
 - smi_ospf6.h, [96](#)
- smi_ospf6_get_nbr_address
 - smi_ospf6.h, [97](#)
- smi_ospf6_get_nbr_address_type
 - smi_ospf6.h, [97](#)
- smi_ospf6_get_nbr_events
 - smi_ospf6.h, [98](#)
- smi_ospf6_get_nbr_hello_suppressed
 - smi_ospf6.h, [98](#)
- smi_ospf6_get_nbr_if_id
 - smi_ospf6.h, [99](#)
- smi_ospf6_get_nbr_lsretransq_len
 - smi_ospf6.h, [99](#)
- smi_ospf6_get_nbr_options
 - smi_ospf6.h, [100](#)
- smi_ospf6_get_nbr_priority
 - smi_ospf6.h, [100](#)
- smi_ospf6_get_nbr_restart_helper_age
 - smi_ospf6.h, [100](#)
- smi_ospf6_get_nbr_restart_helper_exit - reason
 - smi_ospf6.h, [101](#)
- smi_ospf6_get_nbr_restart_helper_status
 - smi_ospf6.h, [101](#)

- smi_ospf6_get_nbr_state
smi_ospf6.h, [102](#)
- smi_ospf6_get_notification_enable
smi_ospf6.h, [102](#)
- smi_ospf6_get_originate_new_lsas
smi_ospf6.h, [103](#)
- smi_ospf6_get_reference_bandwidth
smi_ospf6.h, [103](#)
- smi_ospf6_get_restart_age
smi_ospf6.h, [103](#)
- smi_ospf6_get_restart_exit_reason
smi_ospf6.h, [104](#)
- smi_ospf6_get_restart_interval
smi_ospf6.h, [104](#)
- smi_ospf6_get_restart_status
smi_ospf6.h, [105](#)
- smi_ospf6_get_restart_strict_lsa_check
smi_ospf6.h, [105](#)
- smi_ospf6_get_restart_support
smi_ospf6.h, [105](#)
- smi_ospf6_get_restart_time
smi_ospf6.h, [106](#)
- smi_ospf6_get_router_id
smi_ospf6.h, [106](#)
- smi_ospf6_get_rx_new_lsas
smi_ospf6.h, [106](#)
- smi_ospf6_get_spf_runs
smi_ospf6.h, [107](#)
- smi_ospf6_get_stub_metric
smi_ospf6.h, [107](#)
- smi_ospf6_get_stub_router_-
advertisement
smi_ospf6.h, [108](#)
- smi_ospf6_get_stub_router_support
smi_ospf6.h, [108](#)
- smi_ospf6_get_version_num
smi_ospf6.h, [108](#)
- smi_ospf6_get_virt_if_events
smi_ospf6.h, [109](#)
- smi_ospf6_get_virt_if_hello_interval
smi_ospf6.h, [109](#)
- smi_ospf6_get_virt_if_index
smi_ospf6.h, [109](#)
- smi_ospf6_get_virt_if_instd
smi_ospf6.h, [110](#)
- smi_ospf6_get_virt_if_link_lsa_-
checksumsum
smi_ospf6.h, [110](#)
- smi_ospf6_get_virt_if_link_scope_lsa_-
count
smi_ospf6.h, [111](#)
- smi_ospf6_get_virt_if_retrans_interval
smi_ospf6.h, [111](#)
- smi_ospf6_get_virt_if_rtr_dead_interval
smi_ospf6.h, [112](#)
- smi_ospf6_get_virt_if_state
smi_ospf6.h, [112](#)
- smi_ospf6_get_virt_if_status
smi_ospf6.h, [113](#)
- smi_ospf6_get_virt_if_transmit_delay
smi_ospf6.h, [113](#)
- smi_ospf6_get_virt_link_lsdb_-
advertisement
smi_ospf6.h, [113](#)
- smi_ospf6_get_virt_link_lsdb_age
smi_ospf6.h, [114](#)
- smi_ospf6_get_virt_link_lsdb_checksum
smi_ospf6.h, [114](#)
- smi_ospf6_get_virt_link_lsdb_sequence
smi_ospf6.h, [115](#)
- smi_ospf6_get_virt_link_lsdb_type_-
known
smi_ospf6.h, [116](#)
- smi_ospf6_get_virt_nbr_address
smi_ospf6.h, [116](#)
- smi_ospf6_get_virt_nbr_address_type
smi_ospf6.h, [117](#)
- smi_ospf6_get_virt_nbr_events
smi_ospf6.h, [117](#)
- smi_ospf6_get_virt_nbr_hello_-
suppressed
smi_ospf6.h, [118](#)
- smi_ospf6_get_virt_nbr_if_id
smi_ospf6.h, [118](#)
- smi_ospf6_get_virt_nbr_ifindex
smi_ospf6.h, [119](#)
- smi_ospf6_get_virt_nbr_ifinstd
smi_ospf6.h, [119](#)
- smi_ospf6_get_virt_nbr_ls_retransq_len
smi_ospf6.h, [120](#)
- smi_ospf6_get_virt_nbr_options
smi_ospf6.h, [120](#)
- smi_ospf6_get_virt_nbr_restart_helper_-
age
smi_ospf6.h, [121](#)
- smi_ospf6_get_virt_nbr_restart_helper_-
exit_reason
smi_ospf6.h, [121](#)
- smi_ospf6_get_virt_nbr_restart_helper_-
status

- smi_ospf6.h, [122](#)
- smi_ospf6_get_virt_nbr_state
 - smi_ospf6.h, [122](#)
- smi_ospf6_graceful_restart_set
 - smi_ospf6.h, [123](#)
- smi_ospf6_graceful_restart_unset
 - smi_ospf6.h, [123](#)
- smi_ospf6_if_bfd_disable_set
 - smi_ospf6_bfd.h, [192](#)
- smi_ospf6_if_bfd_disable_unset
 - smi_ospf6_bfd.h, [192](#)
- smi_ospf6_if_bfd_set
 - smi_ospf6_bfd.h, [193](#)
- smi_ospf6_if_bfd_unset
 - smi_ospf6_bfd.h, [193](#)
- smi_ospf6_if_cost_set
 - smi_ospf6.h, [123](#)
- smi_ospf6_if_cost_unset
 - smi_ospf6.h, [124](#)
- smi_ospf6_if_dead_interval_set
 - smi_ospf6.h, [124](#)
- smi_ospf6_if_dead_interval_unset
 - smi_ospf6.h, [125](#)
- smi_ospf6_if_disable_all_set
 - smi_ospf6.h, [125](#)
- smi_ospf6_if_disable_all_unset
 - smi_ospf6.h, [125](#)
- smi_ospf6_if_hello_interval_set
 - smi_ospf6.h, [126](#)
- smi_ospf6_if_hello_interval_unset
 - smi_ospf6.h, [126](#)
- smi_ospf6_if_ipv6_router_set
 - smi_ospf6.h, [127](#)
- smi_ospf6_if_ipv6_router_unset
 - smi_ospf6.h, [127](#)
- smi_ospf6_if_link_lsa_suppression_set
 - smi_ospf6.h, [128](#)
- smi_ospf6_if_mtu_ignore_set
 - smi_ospf6.h, [129](#)
- smi_ospf6_if_mtu_ignore_unset
 - smi_ospf6.h, [129](#)
- smi_ospf6_if_network_set
 - smi_ospf6.h, [130](#)
- smi_ospf6_if_network_unset
 - smi_ospf6.h, [130](#)
- smi_ospf6_if_priority_set
 - smi_ospf6.h, [130](#)
- smi_ospf6_if_priority_unset
 - smi_ospf6.h, [131](#)
- smi_ospf6_if_retransmit_interval_set
 - smi_ospf6.h, [131](#)
- smi_ospf6_if_retransmit_interval_unset
 - smi_ospf6.h, [132](#)
- smi_ospf6_if_te_metric_set
 - smi_ospf6.h, [132](#)
- smi_ospf6_if_te_metric_unset
 - smi_ospf6.h, [133](#)
- smi_ospf6_if_transmit_delay_set
 - smi_ospf6.h, [133](#)
- smi_ospf6_if_transmit_delay_unset
 - smi_ospf6.h, [134](#)
- smi_ospf6_ipv6_ospf_display_route_-single_line_set
 - smi_ospf6.h, [134](#)
- smi_ospf6_ipv6_ospf_display_route_-single_line_unset
 - smi_ospf6.h, [134](#)
- smi_ospf6_log_adjacency_changes_set
 - smi_ospf6.h, [135](#)
- smi_ospf6_log_adjacency_changes_-unset
 - smi_ospf6.h, [135](#)
- smi_ospf6_max_concurrent_dd_set
 - smi_ospf6.h, [136](#)
- smi_ospf6_max_concurrent_dd_unset
 - smi_ospf6.h, [136](#)
- smi_ospf6_max_unuse_lsa_set
 - smi_ospf6.h, [136](#)
- smi_ospf6_max_unuse_lsa_unset
 - smi_ospf6.h, [137](#)
- smi_ospf6_max_unuse_packet_set
 - smi_ospf6.h, [137](#)
- smi_ospf6_max_unuse_packet_unset
 - smi_ospf6.h, [138](#)
- smi_ospf6_no_debug
 - smi_ospf6.h, [138](#)
- smi_ospf6_passive_if_default_set
 - smi_ospf6.h, [140](#)
- smi_ospf6_passive_if_default_unset
 - smi_ospf6.h, [140](#)
- smi_ospf6_passive_if_set
 - smi_ospf6.h, [140](#)
- smi_ospf6_passive_if_unset
 - smi_ospf6.h, [141](#)
- smi_ospf6_process_shutdown_set
 - smi_ospf6.h, [141](#)
- smi_ospf6_process_shutdown_unset
 - smi_ospf6.h, [141](#)
- smi_ospf6_redist_default_set
 - smi_ospf6.h, [142](#)

- smi_ospf6_redist_default_unset
smi_ospf6.h, [142](#)
- smi_ospf6_redistribute_metric_set_by_ospf6_src_tag_sdkapi
smi_ospf6.h, [143](#)
- smi_ospf6_redistribute_metric_set_by_ospf_src_tag_sdkapi
smi_ospf6.h, [143](#)
- smi_ospf6_redistribute_metric_type_set_by_ospf6_src_tag_sdkapi
smi_ospf6.h, [144](#)
- smi_ospf6_redistribute_metric_type_set_by_ospf_src_tag_sdkapi
smi_ospf6.h, [145](#)
- smi_ospf6_redistribute_metric_type_unset_by_ospf6_src_tag_sdkapi
smi_ospf6.h, [145](#)
- smi_ospf6_redistribute_metric_type_unset_by_ospf_src_tag_sdkapi
smi_ospf6.h, [146](#)
- smi_ospf6_redistribute_metric_unset_by_ospf6_src_tag_sdkapi
smi_ospf6.h, [147](#)
- smi_ospf6_redistribute_metric_unset_by_ospf_src_tag_sdkapi
smi_ospf6.h, [147](#)
- smi_ospf6_redistribute_route_tag_set_by_ospf6_src_tag_sdkapi
smi_ospf6.h, [148](#)
- smi_ospf6_redistribute_route_tag_set_by_ospf_src_tag_sdkapi
smi_ospf6.h, [149](#)
- smi_ospf6_redistribute_route_tag_unset_by_ospf6_src_tag_sdkapi
smi_ospf6.h, [149](#)
- smi_ospf6_redistribute_route_tag_unset_by_ospf_src_tag_sdkapi
smi_ospf6.h, [150](#)
- smi_ospf6_redistribute_set_by_ospf6_src_tag_sdkapi
smi_ospf6.h, [151](#)
- smi_ospf6_redistribute_set_by_ospf_src_tag_sdkapi
smi_ospf6.h, [151](#)
- smi_ospf6_redistribute_unset_by_ospf6_src_tag_sdkapi
smi_ospf6.h, [152](#)
- smi_ospf6_redistribute_unset_by_ospf_src_tag_sdkapi
smi_ospf6.h, [152](#)
- smi_ospf6_restart_graceful_set
smi_ospf6.h, [153](#)
- smi_ospf6_restart_helper_grace_period_set
smi_ospf6.h, [154](#)
- smi_ospf6_restart_helper_grace_period_unset
smi_ospf6.h, [154](#)
- smi_ospf6_restart_helper_never_router_id_set
smi_ospf6.h, [154](#)
- smi_ospf6_restart_helper_never_router_id_unset
smi_ospf6.h, [155](#)
- smi_ospf6_restart_helper_never_router_unset_all
smi_ospf6.h, [155](#)
- smi_ospf6_restart_helper_policy_set
smi_ospf6.h, [155](#)
- smi_ospf6_restart_helper_policy_unset
smi_ospf6.h, [156](#)
- smi_ospf6_restart_helper_policy_unset_all
smi_ospf6.h, [156](#)
- smi_ospf6_routemap_default_set
smi_ospf6.h, [157](#)
- smi_ospf6_routemap_default_unset
smi_ospf6.h, [157](#)
- smi_ospf6_routemap_set_by_ospf6_src_tag_sdkapi
smi_ospf6.h, [157](#)
- smi_ospf6_routemap_set_by_ospf_src_tag_sdkapi
smi_ospf6.h, [158](#)
- smi_ospf6_routemap_unset_by_ospf6_src_tag_sdkapi
smi_ospf6.h, [159](#)
- smi_ospf6_routemap_unset_by_ospf_src_tag_sdkapi
smi_ospf6.h, [159](#)
- smi_ospf6_router_id_set
smi_ospf6.h, [160](#)
- smi_ospf6_router_id_unset
smi_ospf6.h, [160](#)
- smi_ospf6_router_set
smi_ospf6.h, [161](#)
- smi_ospf6_router_unset

- smi_ospf6.h, [161](#)
- smi_ospf6_set_area_nssa_trans_role
 - smi_ospf6.h, [161](#)
- smi_ospf6_set_area_stub_metric_type
 - smi_ospf6.h, [162](#)
- smi_ospf6_set_area_te_enabled
 - smi_ospf6.h, [163](#)
- smi_ospf6_set_asbrdr_rtr_status
 - smi_ospf6.h, [163](#)
- smi_ospf6_set_if_admin_stat
 - smi_ospf6.h, [163](#)
- smi_ospf6_set_if_rtr_dead_interval
 - smi_ospf6.h, [164](#)
- smi_ospf6_set_if_type
 - smi_ospf6.h, [165](#)
- smi_ospf6_show_border_routers
 - smi_ospf6.h, [165](#)
- smi_ospf6_show_database
 - smi_ospf6.h, [166](#)
- smi_ospf6_show_debug
 - smi_ospf6.h, [166](#)
- smi_ospf6_show_interface
 - smi_ospf6.h, [166](#)
- smi_ospf6_show_interface_brief
 - smi_ospf6.h, [167](#)
- smi_ospf6_show_interface_brief_by_tag
 - smi_ospf6.h, [167](#)
- smi_ospf6_show_interface_by_tag
 - smi_ospf6.h, [168](#)
- smi_ospf6_show_neighbor_by_if_nbr_id
 - smi_ospf6.h, [168](#)
- smi_ospf6_show_neighbor_by_-
 - interface_name
 - smi_ospf6.h, [169](#)
- smi_ospf6_show_neighbor_by_nbr_id
 - smi_ospf6.h, [169](#)
- smi_ospf6_show_neighbor_detail_by_-
 - if_nbr_id
 - smi_ospf6.h, [170](#)
- smi_ospf6_show_neighbor_detail_by_-
 - interface_name
 - smi_ospf6.h, [170](#)
- smi_ospf6_show_neighbor_detail_by_-
 - tag
 - smi_ospf6.h, [171](#)
- smi_ospf6_show_neighbor_summary_-
 - by_if_nbr_id
 - smi_ospf6.h, [171](#)
- smi_ospf6_show_neighbor_summary_-
 - by_interface_name
 - smi_ospf6.h, [172](#)
- smi_ospf6_show_neighbor_summary_-
 - by_nbr_id
 - smi_ospf6.h, [172](#)
- smi_ospf6_show_neighbor_summary_-
 - by_tag
 - smi_ospf6.h, [173](#)
- smi_ospf6_show_process
 - smi_ospf6.h, [173](#)
- smi_ospf6_show_route
 - smi_ospf6.h, [174](#)
- smi_ospf6_show_route_by_prefix
 - smi_ospf6.h, [174](#)
- smi_ospf6_show_route_summary
 - smi_ospf6.h, [175](#)
- smi_ospf6_show_virtual_link
 - smi_ospf6.h, [175](#)
- smi_ospf6_show_virtual_link_brief
 - smi_ospf6.h, [176](#)
- smi_ospf6_summary_address_ipv4_set_-
 - sdkapi
 - smi_ospf6.h, [176](#)
- smi_ospf6_summary_address_ipv4_-
 - unset_sdkapi
 - smi_ospf6.h, [177](#)
- smi_ospf6_summary_address_not_-
 - advertise_ipv4_set_sdkapi
 - smi_ospf6.h, [178](#)
- smi_ospf6_summary_address_not_-
 - advertise_ipv4_unset_sdkapi
 - smi_ospf6.h, [179](#)
- smi_ospf6_summary_address_not_-
 - advertise_ipv6_unset_sdkapi
 - smi_ospf6.h, [179](#)
- smi_ospf6_summary_address_tag_ipv4_-
 - set_sdkapi
 - smi_ospf6.h, [180](#)
- smi_ospf6_summary_address_tag_ipv4_-
 - unset_sdkapi
 - smi_ospf6.h, [181](#)
- smi_ospf6_timers_spf_set
 - smi_ospf6.h, [182](#)
- smi_ospf6_timers_spf_unset
 - smi_ospf6.h, [182](#)
- smi_ospf6_vlink_bfd_all_set_sdkapi
 - smi_ospf6_bfd.h, [193](#)
- smi_ospf6_vlink_bfd_all_unset_sdkapi
 - smi_ospf6_bfd.h, [194](#)
- smi_ospf6_vlink_bfd_set
 - smi_ospf6_bfd.h, [194](#)

smi_ospf6_vlink_bfd_unset
 smi_ospf6_bfd.h, [194](#)
smi_ospf6_vlink_dead_interval_set
 smi_ospf6.h, [183](#)
smi_ospf6_vlink_dead_interval_unset
 smi_ospf6.h, [183](#)
smi_ospf6_vlink_hello_interval_set
 smi_ospf6.h, [184](#)
smi_ospf6_vlink_hello_interval_unset
 smi_ospf6.h, [184](#)
smi_ospf6_vlink_instance_id_set
 smi_ospf6.h, [185](#)
smi_ospf6_vlink_instance_id_unset
 smi_ospf6.h, [185](#)
smi_ospf6_vlink_retransmit_interval_set
 smi_ospf6.h, [186](#)
smi_ospf6_vlink_retransmit_interval_-
 unset
 smi_ospf6.h, [186](#)
smi_ospf6_vlink_set
 smi_ospf6.h, [187](#)
smi_ospf6_vlink_transmit_delay_set
 smi_ospf6.h, [187](#)
smi_ospf6_vlink_transmit_delay_unset
 smi_ospf6.h, [188](#)
smi_ospf6_vlink_unset
 smi_ospf6.h, [188](#)