

**ZebOS-XP OSPF SMI Reference**  
IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:44 2015



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
2.1	smi_ospf.h File Reference . . . . .	3
2.1.1	Detailed Description . . . . .	70
2.1.2	Function Documentation . . . . .	71
2.1.2.1	smi_debug_ospf_packet_set . . . . .	71
2.1.2.2	smi_debug_ospf_packet_unset . . . . .	71
2.1.2.3	smi_ospf_abr_type_set . . . . .	72
2.1.2.4	smi_ospf_abr_type_unset . . . . .	73
2.1.2.5	smi_ospf_area_auth_by_type_unset . . . . .	73
2.1.2.6	smi_ospf_area_auth_type_set . . . . .	74
2.1.2.7	smi_ospf_area_auth_type_unset . . . . .	74
2.1.2.8	smi_ospf_area_default_cost_set . . . . .	75
2.1.2.9	smi_ospf_area_default_cost_unset . . . . .	75
2.1.2.10	smi_ospf_area_default_cost_value_unset . . . . .	76
2.1.2.11	smi_ospf_area_export_list_set . . . . .	76
2.1.2.12	smi_ospf_area_export_list_unset . . . . .	77
2.1.2.13	smi_ospf_area_filter_list_access_set . . . . .	77
2.1.2.14	smi_ospf_area_filter_list_access_unset . . . . .	78
2.1.2.15	smi_ospf_area_filter_list_prefix_set . . . . .	78
2.1.2.16	smi_ospf_area_filter_list_prefix_unset . . . . .	79
2.1.2.17	smi_ospf_area_import_list_set . . . . .	79
2.1.2.18	smi_ospf_area_import_list_unset . . . . .	80
2.1.2.19	smi_ospf_area_no_summary_set . . . . .	80

2.1.2.20	<a href="#">smi_ospf_area_no_summary_unset</a> . . . . .	80
2.1.2.21	<a href="#">smi_ospf_area_nssa_default_originate_metric_set</a> . . . . .	81
2.1.2.22	<a href="#">smi_ospf_area_nssa_default_originate_metric_- type_set</a> . . . . .	82
2.1.2.23	<a href="#">smi_ospf_area_nssa_default_originate_route_- map_set</a> . . . . .	82
2.1.2.24	<a href="#">smi_ospf_area_nssa_default_originate_set</a> . . . . .	83
2.1.2.25	<a href="#">smi_ospf_area_nssa_default_originate_unset</a> . . . . .	83
2.1.2.26	<a href="#">smi_ospf_area_nssa_no_redistribution_set</a> . . . . .	84
2.1.2.27	<a href="#">smi_ospf_area_nssa_no_redistribution_unset</a> . . . . .	84
2.1.2.28	<a href="#">smi_ospf_area_nssa_set</a> . . . . .	85
2.1.2.29	<a href="#">smi_ospf_area_nssa_stability_interval_set</a> . . . . .	85
2.1.2.30	<a href="#">smi_ospf_area_nssa_translator_role_set</a> . . . . .	86
2.1.2.31	<a href="#">smi_ospf_area_nssa_translator_role_unset</a> . . . . .	86
2.1.2.32	<a href="#">smi_ospf_area_nssa_unset</a> . . . . .	87
2.1.2.33	<a href="#">smi_ospf_area_range_not_advertise_set</a> . . . . .	87
2.1.2.34	<a href="#">smi_ospf_area_range_not_advertise_unset</a> . . . . .	88
2.1.2.35	<a href="#">smi_ospf_area_range_set</a> . . . . .	89
2.1.2.36	<a href="#">smi_ospf_area_range_substitute_set</a> . . . . .	89
2.1.2.37	<a href="#">smi_ospf_area_range_substitute_unset</a> . . . . .	90
2.1.2.38	<a href="#">smi_ospf_area_range_unset</a> . . . . .	90
2.1.2.39	<a href="#">smi_ospf_area_shortcut_set</a> . . . . .	91
2.1.2.40	<a href="#">smi_ospf_area_shortcut_unset</a> . . . . .	91
2.1.2.41	<a href="#">smi_ospf_area_stub_set</a> . . . . .	92
2.1.2.42	<a href="#">smi_ospf_area_stub_unset</a> . . . . .	92
2.1.2.43	<a href="#">smi_ospf_auto_cost_reference_bandwidth_set</a> . . . . .	93
2.1.2.44	<a href="#">smi_ospf_auto_cost_reference_bandwidth_type_set</a> . . . . .	93
2.1.2.45	<a href="#">smi_ospf_auto_cost_reference_bandwidth_unset</a> . . . . .	94
2.1.2.46	<a href="#">smi_ospf_capability_cspf_set</a> . . . . .	94
2.1.2.47	<a href="#">smi_ospf_capability_cspf_unset</a> . . . . .	95
2.1.2.48	<a href="#">smi_ospf_capability_opaque_lsa_set</a> . . . . .	95
2.1.2.49	<a href="#">smi_ospf_capability_opaque_lsa_unset</a> . . . . .	96
2.1.2.50	<a href="#">smi_ospf_capability_restart_set</a> . . . . .	96
2.1.2.51	<a href="#">smi_ospf_capability_restart_unset</a> . . . . .	97

2.1.2.52	<a href="#">smi_ospf_capability_traffic_engineering_set</a>	97
2.1.2.53	<a href="#">smi_ospf_capability_traffic_engineering_unset</a>	97
2.1.2.54	<a href="#">smi_ospf_compatible_rfc1583_set</a>	98
2.1.2.55	<a href="#">smi_ospf_compatible_rfc1583_unset</a>	98
2.1.2.56	<a href="#">smi_ospf_cspf_better_protection_type</a>	99
2.1.2.57	<a href="#">smi_ospf_debug_set</a>	99
2.1.2.58	<a href="#">smi_ospf_debug_unset</a>	101
2.1.2.59	<a href="#">smi_ospf_default_metric_set</a>	102
2.1.2.60	<a href="#">smi_ospf_default_metric_unset</a>	103
2.1.2.61	<a href="#">smi_ospf_disable_db_summary_opt</a>	103
2.1.2.62	<a href="#">smi_ospf_disable_ext_multi_inst</a>	104
2.1.2.63	<a href="#">smi_ospf_distance_all_set</a>	104
2.1.2.64	<a href="#">smi_ospf_distance_all_unset</a>	104
2.1.2.65	<a href="#">smi_ospf_distance_external_set</a>	105
2.1.2.66	<a href="#">smi_ospf_distance_external_unset</a>	105
2.1.2.67	<a href="#">smi_ospf_distance_inter_area_set</a>	106
2.1.2.68	<a href="#">smi_ospf_distance_inter_area_unset</a>	106
2.1.2.69	<a href="#">smi_ospf_distance_intra_area_set</a>	107
2.1.2.70	<a href="#">smi_ospf_distance_intra_area_unset</a>	107
2.1.2.71	<a href="#">smi_ospf_distance_source_set</a>	108
2.1.2.72	<a href="#">smi_ospf_distance_source_unset</a>	108
2.1.2.73	<a href="#">smi_ospf_distribute_list_in_set</a>	109
2.1.2.74	<a href="#">smi_ospf_distribute_list_in_unset</a>	109
2.1.2.75	<a href="#">smi_ospf_distribute_list_out_set</a>	110
2.1.2.76	<a href="#">smi_ospf_distribute_list_out_unset</a>	110
2.1.2.77	<a href="#">smi_ospf_dna_set_sdkapi</a>	111
2.1.2.78	<a href="#">smi_ospf_dna_unset_sdkapi</a>	111
2.1.2.79	<a href="#">smi_ospf_domain_id_set</a>	111
2.1.2.80	<a href="#">smi_ospf_domain_id_unset</a>	112
2.1.2.81	<a href="#">smi_ospf_enable_db_summary_opt</a>	112
2.1.2.82	<a href="#">smi_ospf_enable_ext_multi_inst</a>	113
2.1.2.83	<a href="#">smi_ospf_get_address_less_if</a>	113
2.1.2.84	<a href="#">smi_ospf_get_admin_stat</a>	114
2.1.2.85	<a href="#">smi_ospf_get_area_aggregate_area_id</a>	114

2.1.2.86	smi_ospf_get_area_aggregate_effect . . . . .	115
2.1.2.87	smi_ospf_get_area_aggregate_lsdb_type . . . . .	115
2.1.2.88	smi_ospf_get_area_aggregate_mask . . . . .	116
2.1.2.89	smi_ospf_get_area_aggregate_net . . . . .	116
2.1.2.90	smi_ospf_get_area_aggregate_route_tag . . . . .	117
2.1.2.91	smi_ospf_get_area_aggregate_status . . . . .	117
2.1.2.92	smi_ospf_get_area_bdr_rtr_count . . . . .	118
2.1.2.93	smi_ospf_get_area_bdr_rtr_status . . . . .	118
2.1.2.94	smi_ospf_get_area_id . . . . .	119
2.1.2.95	smi_ospf_get_area_lsa_cksum_sum . . . . .	119
2.1.2.96	smi_ospf_get_area_lsa_count . . . . .	120
2.1.2.97	smi_ospf_get_area_lsa_count_number . . . . .	120
2.1.2.98	smi_ospf_get_area_nssa_translator_events . . . . .	121
2.1.2.99	smi_ospf_get_area_nssa_translator_role . . . . .	121
2.1.2.100	smi_ospf_get_area_nssa_translator_stability_interval . . . . .	121
2.1.2.101	smi_ospf_get_area_nssa_translator_state . . . . .	122
2.1.2.102	smi_ospf_get_area_range_area_id . . . . .	122
2.1.2.103	smi_ospf_get_area_range_effect . . . . .	123
2.1.2.104	smi_ospf_get_area_range_mask . . . . .	123
2.1.2.105	smi_ospf_get_area_range_net . . . . .	124
2.1.2.106	smi_ospf_get_area_range_status . . . . .	124
2.1.2.107	smi_ospf_get_area_status . . . . .	125
2.1.2.108	smi_ospf_get_area_summary . . . . .	125
2.1.2.109	smi_ospf_get_as_lsdb_age . . . . .	126
2.1.2.110	smi_ospf_get_as_lsdb_checksum . . . . .	126
2.1.2.111	smi_ospf_get_as_lsdb_sequence . . . . .	127
2.1.2.112	smi_ospf_get_as_scope_lsa_count . . . . .	127
2.1.2.113	smi_ospf_get_asbdr_rtr_count . . . . .	128
2.1.2.114	smi_ospf_get_asbdr_rtr_status . . . . .	128
2.1.2.115	smi_ospf_get_auth_type . . . . .	128
2.1.2.116	smi_ospf_get_compatible_rfc1583 . . . . .	129
2.1.2.117	smi_ospf_get_demand_extensions . . . . .	129
2.1.2.118	smi_ospf_get_discontinuity_time . . . . .	130
2.1.2.119	smi_ospf_get_domain_id . . . . .	130

2.1.2.120 smi_ospf_get_exit_overflow_interval . . . . .	130
2.1.2.121 smi_ospf_get_ext_lsdb_advertisement . . . . .	131
2.1.2.122 smi_ospf_get_ext_lsdb_age . . . . .	131
2.1.2.123 smi_ospf_get_ext_lsdb_checksum . . . . .	132
2.1.2.124 smi_ospf_get_ext_lsdb_limit . . . . .	132
2.1.2.125 smi_ospf_get_ext_lsdb_lsid . . . . .	133
2.1.2.126 smi_ospf_get_ext_lsdb_router_id . . . . .	134
2.1.2.127 smi_ospf_get_ext_lsdb_sequence . . . . .	134
2.1.2.128 smi_ospf_get_ext_lsdb_type . . . . .	135
2.1.2.129 smi_ospf_get_extern_lsa_cksum_sum . . . . .	135
2.1.2.130 smi_ospf_get_extern_lsa_count . . . . .	136
2.1.2.131 smi_ospf_get_external_type1_metric . . . . .	136
2.1.2.132 smi_ospf_get_external_type2_metric . . . . .	136
2.1.2.133 smi_ospf_get_host_area_id . . . . .	137
2.1.2.134 smi_ospf_get_host_cfg_area_id . . . . .	137
2.1.2.135 smi_ospf_get_host_ip_address . . . . .	138
2.1.2.136 smi_ospf_get_host_metric . . . . .	138
2.1.2.137 smi_ospf_get_host_status . . . . .	138
2.1.2.138 smi_ospf_get_host_tos . . . . .	139
2.1.2.139 smi_ospf_get_if_admin_stat . . . . .	139
2.1.2.140 smi_ospf_get_if_area_id . . . . .	140
2.1.2.141 smi_ospf_get_if_auth_key . . . . .	140
2.1.2.142 smi_ospf_get_if_auth_type . . . . .	141
2.1.2.143 smi_ospf_get_if_backup_designated_router . . . . .	141
2.1.2.144 smi_ospf_get_if_bdr . . . . .	142
2.1.2.145 smi_ospf_get_if_demand . . . . .	142
2.1.2.146 smi_ospf_get_if_designated_router . . . . .	143
2.1.2.147 smi_ospf_get_if_dr . . . . .	143
2.1.2.148 smi_ospf_get_if_events . . . . .	144
2.1.2.149 smi_ospf_get_if_hello_interval . . . . .	144
2.1.2.150 smi_ospf_get_if_ip_address . . . . .	145
2.1.2.151 smi_ospf_get_if_lsa_checksum . . . . .	145
2.1.2.152 smi_ospf_get_if_lsa_count . . . . .	146
2.1.2.153 smi_ospf_get_if_metric_address_less_if . . . . .	146

2.1.2.154 smi_ospf_get_if_metric_ip_address . . . . .	147
2.1.2.155 smi_ospf_get_if_metric_status . . . . .	147
2.1.2.156 smi_ospf_get_if_metric_value . . . . .	148
2.1.2.157 smi_ospf_get_if_multicast_forwarding . . . . .	148
2.1.2.158 smi_ospf_get_if_poll_interval . . . . .	149
2.1.2.159 smi_ospf_get_if_retrans_interval . . . . .	149
2.1.2.160 smi_ospf_get_if_rtr_dead_interval . . . . .	150
2.1.2.161 smi_ospf_get_if_rtr_priority . . . . .	150
2.1.2.162 smi_ospf_get_if_state . . . . .	151
2.1.2.163 smi_ospf_get_if_status . . . . .	151
2.1.2.164 smi_ospf_get_if_transit_delay . . . . .	152
2.1.2.165 smi_ospf_get_if_type . . . . .	152
2.1.2.166 smi_ospf_get_import_as_extern . . . . .	153
2.1.2.167 smi_ospf_get_inter_area_metric . . . . .	153
2.1.2.168 smi_ospf_get_intra_area_metric . . . . .	154
2.1.2.169 smi_ospf_get_local_lsdb_age . . . . .	154
2.1.2.170 smi_ospf_get_local_lsdb_checksum . . . . .	154
2.1.2.171 smi_ospf_get_local_lsdb_sequence . . . . .	155
2.1.2.172 smi_ospf_get_lsdb_advertisement . . . . .	156
2.1.2.173 smi_ospf_get_lsdb_age . . . . .	156
2.1.2.174 smi_ospf_get_lsdb_area_id . . . . .	157
2.1.2.175 smi_ospf_get_lsdb_checksum . . . . .	157
2.1.2.176 smi_ospf_get_lsdb_lsid . . . . .	158
2.1.2.177 smi_ospf_get_lsdb_router_id . . . . .	158
2.1.2.178 smi_ospf_get_lsdb_sequence . . . . .	159
2.1.2.179 smi_ospf_get_lsdb_type . . . . .	159
2.1.2.180 smi_ospf_get_nbma_nbr_permanence . . . . .	160
2.1.2.181 smi_ospf_get_nbma_nbr_status . . . . .	161
2.1.2.182 smi_ospf_get_nbr_address_less_index . . . . .	161
2.1.2.183 smi_ospf_get_nbr_events . . . . .	162
2.1.2.184 smi_ospf_get_nbr_hello_suppressed . . . . .	162
2.1.2.185 smi_ospf_get_nbr_ip_addr . . . . .	163
2.1.2.186 smi_ospf_get_nbr_ls_retrans_qlen . . . . .	163
2.1.2.187 smi_ospf_get_nbr_options . . . . .	164



2.1.2.188 smi_ospf_get_nbr_priority . . . . .	164
2.1.2.189 smi_ospf_get_nbr_restart_helper_age . . . . .	165
2.1.2.190 smi_ospf_get_nbr_restart_helper_exit_reason . . . . .	165
2.1.2.191 smi_ospf_get_nbr_restart_helper_status . . . . .	166
2.1.2.192 smi_ospf_get_nbr_rtr_id . . . . .	166
2.1.2.193 smi_ospf_get_nbr_state . . . . .	167
2.1.2.194 smi_ospf_get_opaque_lsa_support . . . . .	167
2.1.2.195 smi_ospf_get_originate_new_lsas . . . . .	168
2.1.2.196 smi_ospf_get_reference_bandwidth . . . . .	168
2.1.2.197 smi_ospf_get_restart_age . . . . .	168
2.1.2.198 smi_ospf_get_restart_exit_reason . . . . .	169
2.1.2.199 smi_ospf_get_restart_interval . . . . .	169
2.1.2.200 smi_ospf_get_restart_status . . . . .	170
2.1.2.201 smi_ospf_get_restart_strict_lsa_check . . . . .	170
2.1.2.202 smi_ospf_get_restart_support . . . . .	170
2.1.2.203 smi_ospf_get_router_id . . . . .	171
2.1.2.204 smi_ospf_get_rx_new_lsas . . . . .	171
2.1.2.205 smi_ospf_get_settrap . . . . .	172
2.1.2.206 smi_ospf_get_spf_runs . . . . .	172
2.1.2.207 smi_ospf_get_stub_area_id . . . . .	172
2.1.2.208 smi_ospf_get_stub_metric . . . . .	173
2.1.2.209 smi_ospf_get_stub_metric_type . . . . .	173
2.1.2.210 smi_ospf_get_stub_router_advertisement . . . . .	174
2.1.2.211 smi_ospf_get_stub_router_support . . . . .	174
2.1.2.212 smi_ospf_get_stub_status . . . . .	175
2.1.2.213 smi_ospf_get_stub_tos . . . . .	175
2.1.2.214 smi_ospf_get_tos_support . . . . .	175
2.1.2.215 smi_ospf_get_version_number . . . . .	176
2.1.2.216 smi_ospf_get_virt_if_area_id . . . . .	176
2.1.2.217 smi_ospf_get_virt_if_auth_key . . . . .	177
2.1.2.218 smi_ospf_get_virt_if_auth_type . . . . .	177
2.1.2.219 smi_ospf_get_virt_if_events . . . . .	178
2.1.2.220 smi_ospf_get_virt_if_hello_interval . . . . .	178
2.1.2.221 smi_ospf_get_virt_if_lsa_cksumsum . . . . .	179

2.1.2.222 smi_ospf_get_virt_if_lsa_count . . . . .	179
2.1.2.223 smi_ospf_get_virt_if_neighbor . . . . .	180
2.1.2.224 smi_ospf_get_virt_if_retrans_interval . . . . .	180
2.1.2.225 smi_ospf_get_virt_if_rtr_dead_interval . . . . .	181
2.1.2.226 smi_ospf_get_virt_if_state . . . . .	181
2.1.2.227 smi_ospf_get_virt_if_status . . . . .	182
2.1.2.228 smi_ospf_get_virt_if_transit_delay . . . . .	182
2.1.2.229 smi_ospf_get_virt_local_lsdb_age . . . . .	183
2.1.2.230 smi_ospf_get_virt_local_lsdb_checksum . . . . .	183
2.1.2.231 smi_ospf_get_virt_local_lsdb_sequence . . . . .	184
2.1.2.232 smi_ospf_get_virt_nbr_area . . . . .	184
2.1.2.233 smi_ospf_get_virt_nbr_events . . . . .	185
2.1.2.234 smi_ospf_get_virt_nbr_hello_suppressed . . . . .	185
2.1.2.235 smi_ospf_get_virt_nbr_ip_addr . . . . .	186
2.1.2.236 smi_ospf_get_virt_nbr_ls_retrans_qlen . . . . .	186
2.1.2.237 smi_ospf_get_virt_nbr_options . . . . .	187
2.1.2.238 smi_ospf_get_virt_nbr_restart_helper_age . . . . .	187
2.1.2.239 smi_ospf_get_virt_nbr_restart_helper_exit_reason . . . . .	188
2.1.2.240 smi_ospf_get_virt_nbr_restart_helper_status . . . . .	188
2.1.2.241 smi_ospf_get_virt_nbr_rtr_id . . . . .	189
2.1.2.242 smi_ospf_get_virt_nbr_state . . . . .	189
2.1.2.243 smi_ospf_graceful_restart_planned_set_sdkapi . . . . .	190
2.1.2.244 smi_ospf_graceful_restart_planned_unset_sdkapi . . . . .	190
2.1.2.245 smi_ospf_graceful_restart_set_sdkapi . . . . .	190
2.1.2.246 smi_ospf_graceful_restart_unset_sdkapi . . . . .	191
2.1.2.247 smi_ospf_host_entry_cost_set . . . . .	191
2.1.2.248 smi_ospf_host_entry_cost_unset . . . . .	191
2.1.2.249 smi_ospf_host_entry_set . . . . .	192
2.1.2.250 smi_ospf_host_entry_unset . . . . .	192
2.1.2.251 smi_ospf_if_authentication_key_set . . . . .	193
2.1.2.252 smi_ospf_if_authentication_key_set_by_addr . . . . .	193
2.1.2.253 smi_ospf_if_authentication_key_unset . . . . .	194
2.1.2.254 smi_ospf_if_authentication_key_unset_by_addr . . . . .	194
2.1.2.255 smi_ospf_if_authentication_type_set . . . . .	195

2.1.2.256 smi_ospf_if_authentication_type_set_by_addr . . .	195
2.1.2.257 smi_ospf_if_authentication_type_unset . . . . .	196
2.1.2.258 smi_ospf_if_authentication_type_unset_by_addr . .	196
2.1.2.259 smi_ospf_if_conf_ldp_igp_set_sdkapi . . . . .	197
2.1.2.260 smi_ospf_if_conf_ldp_igp_unset_sdkapi . . . . .	197
2.1.2.261 smi_ospf_if_cost_set . . . . .	197
2.1.2.262 smi_ospf_if_cost_set_by_addr . . . . .	198
2.1.2.263 smi_ospf_if_cost_unset . . . . .	198
2.1.2.264 smi_ospf_if_cost_unset_by_addr . . . . .	199
2.1.2.265 smi_ospf_if_cost_value_unset . . . . .	199
2.1.2.266 smi_ospf_if_database_filter_set . . . . .	199
2.1.2.267 smi_ospf_if_database_filter_set_by_addr . . . . .	200
2.1.2.268 smi_ospf_if_database_filter_unset . . . . .	200
2.1.2.269 smi_ospf_if_database_filter_unset_by_addr . . . .	201
2.1.2.270 smi_ospf_if_dead_interval_set . . . . .	201
2.1.2.271 smi_ospf_if_dead_interval_set_by_addr . . . . .	201
2.1.2.272 smi_ospf_if_dead_interval_unset . . . . .	202
2.1.2.273 smi_ospf_if_dead_interval_unset_by_addr . . . . .	202
2.1.2.274 smi_ospf_if_disable_all_set . . . . .	203
2.1.2.275 smi_ospf_if_disable_all_unset . . . . .	203
2.1.2.276 smi_ospf_if_dna_set . . . . .	204
2.1.2.277 smi_ospf_if_dna_unset . . . . .	204
2.1.2.278 smi_ospf_if_hello_interval_set . . . . .	205
2.1.2.279 smi_ospf_if_hello_interval_set_by_addr . . . . .	205
2.1.2.280 smi_ospf_if_hello_interval_unset . . . . .	205
2.1.2.281 smi_ospf_if_hello_interval_unset_by_addr . . . . .	206
2.1.2.282 smi_ospf_if_ip_router_set . . . . .	206
2.1.2.283 smi_ospf_if_ip_router_unset . . . . .	207
2.1.2.284 smi_ospf_if_message_digest_key_get . . . . .	207
2.1.2.285 smi_ospf_if_message_digest_key_set . . . . .	208
2.1.2.286 smi_ospf_if_message_digest_key_set_by_addr . . .	208
2.1.2.287 smi_ospf_if_message_digest_key_unset . . . . .	209
2.1.2.288 smi_ospf_if_message_digest_key_unset_by_addr . .	209
2.1.2.289 smi_ospf_if_mtu_ignore_set . . . . .	210

2.1.2.290 smi_ospf_if_mtu_ignore_set_by_addr . . . . .	210
2.1.2.291 smi_ospf_if_mtu_ignore_unset . . . . .	211
2.1.2.292 smi_ospf_if_mtu_ignore_unset_by_addr . . . . .	211
2.1.2.293 smi_ospf_if_mtu_set . . . . .	212
2.1.2.294 smi_ospf_if_mtu_unset . . . . .	212
2.1.2.295 smi_ospf_if_network_p2mp_nbma_set . . . . .	212
2.1.2.296 smi_ospf_if_network_set . . . . .	213
2.1.2.297 smi_ospf_if_network_unset . . . . .	213
2.1.2.298 smi_ospf_if_passive_interface_set . . . . .	214
2.1.2.299 smi_ospf_if_passive_interface_unset . . . . .	214
2.1.2.300 smi_ospf_if_priority_set . . . . .	214
2.1.2.301 smi_ospf_if_priority_set_by_addr . . . . .	215
2.1.2.302 smi_ospf_if_priority_unset . . . . .	215
2.1.2.303 smi_ospf_if_priority_unset_by_addr . . . . .	216
2.1.2.304 smi_ospf_if_resync_timeout_set . . . . .	216
2.1.2.305 smi_ospf_if_resync_timeout_set_by_addr . . . . .	216
2.1.2.306 smi_ospf_if_resync_timeout_unset . . . . .	217
2.1.2.307 smi_ospf_if_resync_timeout_unset_by_addr . . . . .	217
2.1.2.308 smi_ospf_if_retransmit_interval_set . . . . .	218
2.1.2.309 smi_ospf_if_retransmit_interval_set_by_addr . . . . .	218
2.1.2.310 smi_ospf_if_retransmit_interval_unset . . . . .	219
2.1.2.311 smi_ospf_if_retransmit_interval_unset_by_addr . . . . .	219
2.1.2.312 smi_ospf_if_te_metric_set . . . . .	219
2.1.2.313 smi_ospf_if_te_metric_unset . . . . .	220
2.1.2.314 smi_ospf_if_transmit_delay_set . . . . .	220
2.1.2.315 smi_ospf_if_transmit_delay_set_by_addr . . . . .	221
2.1.2.316 smi_ospf_if_transmit_delay_unset . . . . .	221
2.1.2.317 smi_ospf_if_transmit_delay_unset_by_addr . . . . .	222
2.1.2.318 smi_ospf_log_adj_changes_set . . . . .	222
2.1.2.319 smi_ospf_log_adj_changes_unset . . . . .	222
2.1.2.320 smi_ospf_lsa_min_arrival_set . . . . .	223
2.1.2.321 smi_ospf_lsa_min_arrival_unset . . . . .	223
2.1.2.322 smi_ospf_lsa_throttle_timers_set . . . . .	224
2.1.2.323 smi_ospf_lsa_throttle_timers_unset . . . . .	224

2.1.2.324 smi_ospf_max_area_limit_set_sdkapi . . . . .	225
2.1.2.325 smi_ospf_max_area_limit_unset_sdkapi . . . . .	225
2.1.2.326 smi_ospf_max_concurrent_dd_set . . . . .	225
2.1.2.327 smi_ospf_max_concurrent_dd_unset . . . . .	226
2.1.2.328 smi_ospf_max_unuse_lsa_set . . . . .	226
2.1.2.329 smi_ospf_max_unuse_lsa_unset . . . . .	227
2.1.2.330 smi_ospf_max_unuse_packet_set . . . . .	227
2.1.2.331 smi_ospf_max_unuse_packet_unset . . . . .	228
2.1.2.332 smi_ospf_multi_area_adjacency_set . . . . .	228
2.1.2.333 smi_ospf_multi_area_adjacency_unset . . . . .	229
2.1.2.334 smi_ospf_nbr_static_cost_set . . . . .	229
2.1.2.335 smi_ospf_nbr_static_cost_unset . . . . .	230
2.1.2.336 smi_ospf_nbr_static_poll_interval_set . . . . .	230
2.1.2.337 smi_ospf_nbr_static_poll_interval_unset . . . . .	231
2.1.2.338 smi_ospf_nbr_static_priority_set . . . . .	231
2.1.2.339 smi_ospf_nbr_static_priority_unset . . . . .	232
2.1.2.340 smi_ospf_nbr_static_set . . . . .	232
2.1.2.341 smi_ospf_nbr_static_unset . . . . .	233
2.1.2.342 smi_ospf_network_set . . . . .	233
2.1.2.343 smi_ospf_network_unset . . . . .	234
2.1.2.344 smi_ospf_opaque_area_lsa_set . . . . .	234
2.1.2.345 smi_ospf_opaque_as_lsa_set . . . . .	235
2.1.2.346 smi_ospf_opaque_link_lsa_set . . . . .	236
2.1.2.347 smi_ospf_opaque_te_link_local_lsa_disable . . . . .	236
2.1.2.348 smi_ospf_opaque_te_link_local_lsa_enable . . . . .	237
2.1.2.349 smi_ospf_overflow_database_external_interval_set . . . . .	237
2.1.2.350 smi_ospf_overflow_database_external_interval_unset . . . . .	237
2.1.2.351 smi_ospf_overflow_database_external_limit_set . . . . .	238
2.1.2.352 smi_ospf_overflow_database_external_limit_unset . . . . .	238
2.1.2.353 smi_ospf_passive_interface_default_set . . . . .	239
2.1.2.354 smi_ospf_passive_interface_default_unset . . . . .	239
2.1.2.355 smi_ospf_passive_interface_set . . . . .	240
2.1.2.356 smi_ospf_passive_interface_set_by_addr . . . . .	240
2.1.2.357 smi_ospf_passive_interface_unset . . . . .	241

2.1.2.358 smi_ospf_passive_interface_unset_by_addr . . . . .	241
2.1.2.359 smi_ospf_process_set . . . . .	242
2.1.2.360 smi_ospf_process_set_vrf . . . . .	242
2.1.2.361 smi_ospf_process_shut_set . . . . .	242
2.1.2.362 smi_ospf_process_shut_unset . . . . .	243
2.1.2.363 smi_ospf_process_unset . . . . .	243
2.1.2.364 smi_ospf_redist_default_set . . . . .	244
2.1.2.365 smi_ospf_redist_default_unset . . . . .	244
2.1.2.366 smi_ospf_redist_metric_set . . . . .	245
2.1.2.367 smi_ospf_redist_metric_type_set . . . . .	245
2.1.2.368 smi_ospf_redist_metric_type_unset . . . . .	246
2.1.2.369 smi_ospf_redist_metric_unset . . . . .	246
2.1.2.370 smi_ospf_redist_proto_set . . . . .	247
2.1.2.371 smi_ospf_redist_proto_unset . . . . .	247
2.1.2.372 smi_ospf_redist_tag_set . . . . .	248
2.1.2.373 smi_ospf_redist_tag_unset . . . . .	249
2.1.2.374 smi_ospf_redistribute_default_set . . . . .	249
2.1.2.375 smi_ospf_redistribute_set . . . . .	250
2.1.2.376 smi_ospf_restart_graceful_sdkapi . . . . .	250
2.1.2.377 smi_ospf_restart_helper_grace_period_set . . . . .	251
2.1.2.378 smi_ospf_restart_helper_grace_period_unset . . . . .	251
2.1.2.379 smi_ospf_restart_helper_never_router_set . . . . .	251
2.1.2.380 smi_ospf_restart_helper_never_router_unset . . . . .	252
2.1.2.381 smi_ospf_restart_helper_never_router_unset_all . . . . .	252
2.1.2.382 smi_ospf_restart_helper_policy_set . . . . .	253
2.1.2.383 smi_ospf_restart_helper_policy_unset . . . . .	253
2.1.2.384 smi_ospf_routemap_default_set . . . . .	253
2.1.2.385 smi_ospf_routemap_default_unset . . . . .	254
2.1.2.386 smi_ospf_routemap_set . . . . .	254
2.1.2.387 smi_ospf_routemap_unset . . . . .	255
2.1.2.388 smi_ospf_router_id_set . . . . .	255
2.1.2.389 smi_ospf_router_id_unset . . . . .	256
2.1.2.390 smi_ospf_set_area_aggregate_effect . . . . .	256
2.1.2.391 smi_ospf_set_area_aggregate_route_tag . . . . .	257

2.1.2.392 smi_ospf_set_area_aggregate_status . . . . .	257
2.1.2.393 smi_ospf_set_area_status . . . . .	258
2.1.2.394 smi_ospf_set_asbdr_rtr_status . . . . .	258
2.1.2.395 smi_ospf_set_if_admin_stat . . . . .	259
2.1.2.396 smi_ospf_set_if_area_id . . . . .	259
2.1.2.397 smi_ospf_set_if_auth_key . . . . .	260
2.1.2.398 smi_ospf_set_if_auth_type . . . . .	260
2.1.2.399 smi_ospf_set_if_hello_interval . . . . .	261
2.1.2.400 smi_ospf_set_if_metric_status . . . . .	262
2.1.2.401 smi_ospf_set_if_metric_value . . . . .	262
2.1.2.402 smi_ospf_set_if_poll_interval . . . . .	263
2.1.2.403 smi_ospf_set_if_retrans_interval . . . . .	263
2.1.2.404 smi_ospf_set_if_rtr_dead_interval . . . . .	264
2.1.2.405 smi_ospf_set_if_rtr_priority . . . . .	264
2.1.2.406 smi_ospf_set_if_status . . . . .	265
2.1.2.407 smi_ospf_set_if_transit_delay . . . . .	265
2.1.2.408 smi_ospf_set_if_type . . . . .	266
2.1.2.409 smi_ospf_set_import_as_extern . . . . .	267
2.1.2.410 smi_ospf_set_lsdb_limit_sdkapi . . . . .	267
2.1.2.411 smi_ospf_set_multicast_extensions . . . . .	268
2.1.2.412 smi_ospf_set_nbma_nbr_status . . . . .	268
2.1.2.413 smi_ospf_set_nbr_priority . . . . .	269
2.1.2.414 smi_ospf_set_nssa_stability_interval . . . . .	269
2.1.2.415 smi_ospf_set_settrap . . . . .	270
2.1.2.416 smi_ospf_set_virt_if_retrans_interval . . . . .	270
2.1.2.417 smi_ospf_set_virt_if_status . . . . .	271
2.1.2.418 smi_ospf_summary_address_not_advertise_set . . .	271
2.1.2.419 smi_ospf_summary_address_not_advertise_unset .	272
2.1.2.420 smi_ospf_summary_address_set . . . . .	272
2.1.2.421 smi_ospf_summary_address_tag_set . . . . .	273
2.1.2.422 smi_ospf_summary_address_tag_unset . . . . .	273
2.1.2.423 smi_ospf_summary_address_unset . . . . .	274
2.1.2.424 smi_ospf_te_link_detail_set . . . . .	274
2.1.2.425 smi_ospf_te_link_enable_sdkapi . . . . .	275

2.1.2.426 smi_ospf_te_link_flood_scope_set . . . . .	275
2.1.2.427 smi_ospf_te_link_flood_scope_unset . . . . .	276
2.1.2.428 smi_ospf_te_link_set . . . . .	276
2.1.2.429 smi_ospf_telink_te_metric_set . . . . .	277
2.1.2.430 smi_ospf_telink_te_metric_unset . . . . .	277
2.1.2.431 smi_ospf_timers_refresh_set . . . . .	278
2.1.2.432 smi_ospf_timers_refresh_unset . . . . .	278
2.1.2.433 smi_ospf_timers_spf_set . . . . .	279
2.1.2.434 smi_ospf_timers_spf_unset . . . . .	279
2.1.2.435 smi_ospf_timers_spf_validate_and_unset . . . . .	280
2.1.2.436 smi_ospf_vlink_authentication_key_set . . . . .	280
2.1.2.437 smi_ospf_vlink_authentication_key_unset . . . . .	281
2.1.2.438 smi_ospf_vlink_authentication_type_set . . . . .	281
2.1.2.439 smi_ospf_vlink_authentication_type_unset . . . . .	282
2.1.2.440 smi_ospf_vlink_dead_interval_set . . . . .	282
2.1.2.441 smi_ospf_vlink_dead_interval_unset . . . . .	283
2.1.2.442 smi_ospf_vlink_hello_interval_set . . . . .	284
2.1.2.443 smi_ospf_vlink_hello_interval_unset . . . . .	284
2.1.2.444 smi_ospf_vlink_message_digest_key_set . . . . .	285
2.1.2.445 smi_ospf_vlink_message_digest_key_unset . . . . .	285
2.1.2.446 smi_ospf_vlink_retransmit_interval_set . . . . .	286
2.1.2.447 smi_ospf_vlink_retransmit_interval_unset . . . . .	286
2.1.2.448 smi_ospf_vlink_set . . . . .	287
2.1.2.449 smi_ospf_vlink_transmit_delay_set . . . . .	288
2.1.2.450 smi_ospf_vlink_transmit_delay_unset . . . . .	288
2.1.2.451 smi_ospf_vlink_unset . . . . .	289
2.1.2.452 smi_show_if_info . . . . .	289
2.1.2.453 smi_show_ospf_borderrouter_info . . . . .	290
2.1.2.454 smi_show_ospf_buffer_info . . . . .	290
2.1.2.455 smi_show_ospf_database_detail_info . . . . .	290
2.1.2.456 smi_show_ospf_database_summary_info . . . . .	291
2.1.2.457 smi_show_ospf_interface_brief_info . . . . .	291
2.1.2.458 smi_show_ospf_multiarea_info . . . . .	292
2.1.2.459 smi_show_ospf_nbr_info . . . . .	292



---

2.1.2.460	<a href="#">smi_show_ospf_proc_info</a>	293
2.1.2.461	<a href="#">smi_show_ospf_route_info</a>	293
2.1.2.462	<a href="#">smi_show_ospf_route_summary_info</a>	294
2.1.2.463	<a href="#">smi_show_ospf_vlink_info</a>	294
2.2	<a href="#">smi_ospf_bfd.h File Reference</a>	295
2.2.1	<a href="#">Detailed Description</a>	296
2.2.2	<a href="#">Function Documentation</a>	296
2.2.2.1	<a href="#">smi_ospf_bfd_all_interfaces_set</a>	296
2.2.2.2	<a href="#">smi_ospf_bfd_all_interfaces_unset</a>	296
2.2.2.3	<a href="#">smi_ospf_if_bfd_disable_set</a>	297
2.2.2.4	<a href="#">smi_ospf_if_bfd_disable_unset</a>	297
2.2.2.5	<a href="#">smi_ospf_if_bfd_set</a>	298
2.2.2.6	<a href="#">smi_ospf_if_bfd_unset</a>	298
2.2.2.7	<a href="#">smi_ospf_vlink_bfd_set</a>	298
2.2.2.8	<a href="#">smi_ospf_vlink_bfd_unset</a>	299



# Chapter 1

## File Index

### 1.1 File List

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

<a href="#">smi_ospf.h</a> (Provides API for managing OSPF. Open Shortest Path First (OSPF) is an interior gateway protocol that was designed for TCP/IP networks to address the scaling issues of distance-vector routing protocols such as RIP ) . . . . .	3
<a href="#">smi_ospf_bfd.h</a> (Provides APIs for managing Bidirectional Forwarding Detection(BFD) in ZebOS ) . . . . .	295



## Chapter 2

# File Documentation

### 2.1 smi\_ospf.h File Reference

Provides API for managing OSPF. Open Shortest Path First (OSPF) is an interior gateway protocol that was designed for TCP/IP networks to address the scaling issues of distance-vector routing protocols such as RIP. `#include "smi_client.h"`

```
#include "smi_ospf_msg.h"
```

```
#include "smi_ospf_bfd.h"
```

#### Defines

- `#define SMI_OSPF_AREA_ID_FORMAT_ADDRESS 1`
- `#define SMI_OSPF_IEA_ID_FORMAT_DECIMAL 2`

#### Functions

- `int smi_ospf_str_to_area (char *str, struct pal_in4_addr *areaId, int *areaFormat)`
- `int smi_ospf_process_set (struct smiclient_globals *azg, u_int32_t vrId, int ospfProcessId)`

*Sets This function creates an OSPF instance.*

- `int smi_ospf_process_set_vrf (struct smiclient_globals *azg, u_int32_t vrId, int ospfProcessId, char *vrfName)`

*Sets VRF name to an OSPF process.*

- `int smi_ospf_te_link_enable_sdkapi (struct smiclient_globals *azg, u_int32_t vr_id, char *teLinkName)`

*Sets TE link mode to an OSPF process.*

- int [smi\\_ospf\\_process\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*Sets This function destroys the specified OSPF process.*

- int [smi\\_ospf\\_network\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr ospfNetAddr, u\_char netMask, struct pal\_in4\_addr areaId, s\_int16\_t ospfInterfaceInstanceId)

*Sets This function enables an interface for the OSPF domain.*

- int [smi\\_ospf\\_network\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr ospfNetAddr, u\_char netMask, struct pal\_in4\_addr areaId, s\_int16\_t ospfInterfaceInstanceId)

*Sets This function deletes the network area configuration directive matched to a specified prefix and area.*

- int [smi\\_ospf\\_domain\\_id\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*domainType, int domainValue, bool\_t isPrimaryDomainId)

*Sets This function sets an OSPF domain ID as specified: domain ID for a particular OSPF VRF instance.*

- int [smi\\_ospf\\_domain\\_id\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*domainType, u\_int8\_t \*domainValue, bool\_t isPrimaryDomainId)

*Sets This function removes the OSPF domain ID.*

- int [smi\\_ospf\\_null\\_domain\\_id\\_set\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, int proc\_id, bool\_t nullDomainValue)

- int [smi\\_ospf\\_get\\_domain\\_id](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct smi\_ospf\_vrf\_domain\_id \*ret\_pdomain\_id, struct list \*sDomain\_list)

*Get the configured domain\_id information.*

- int [smi\\_ospf\\_router\\_id\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr routerId)

*Sets This function sets the static OSPF router ID to the specified value.*

- int [smi\\_ospf\\_router\\_id\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*Sets This function resets the static OSPF router ID to the default value: 0.*

- int [smi\\_ospf\\_passive\\_interface\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*passiveIfName)

*Sets This function sets the specified interface to passive mode (OSPF\_IF\_PASSIVE).*

- int [smi\\_ospf\\_passive\\_interface\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*passiveIfName, struct pal\_in4\_addr passiveIfIpAddr)

*Sets This function sets the interface specified by IP address to passive mode.*

- int [smi\\_ospf\\_passive\\_interface\\_default\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*Sets This function sets all interfaces to passive mode by default.*

- int [smi\\_ospf\\_passive\\_interface\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*ifName)

*Sets This function resets the current interface to active mode.*

- int [smi\\_ospf\\_passive\\_interface\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*passiveIfName, struct pal\_in4\_addr passiveIfIpAddr)

*Sets This function resets the current interface to active mode.*

- int [smi\\_ospf\\_passive\\_interface\\_default\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*Sets This function resets all interfaces.*

- int [smi\\_ospf\\_host\\_entry\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr hostIpAddr, struct pal\_in4\_addr areaId)

*This sets the ospf stub entry host address.*

- int [smi\\_ospf\\_host\\_entry\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr hostIpAddr, struct pal\_in4\_addr areaId)

*This function removes the ospf stub entry address.*

- int [smi\\_ospf\\_host\\_entry\\_cost\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr hostIpAddr, struct pal\_in4\_addr areaId, int hostEntryCost)

*This function sets stub host entry belonging to particular area along with cost.*

- int [smi\\_ospf\\_abr\\_type\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_char areaBorderType)

*Sets This function sets the OSPF area border route (ABR) type.*

- int [smi\\_ospf\\_abr\\_type\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*Sets This function resets the configured ABR type.*

- int [smi\\_ospf\\_compatible\\_rfc1583\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*Sets This function calculates route summary costs according to RFC 1583.*

- int [smi\\_ospf\\_compatible\\_rfc1583\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*Sets This function disables the calculation of route summary costs according to RFC 1583.*

- `int smi_ospf_timers_spf_set` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t spfMinDelay, u\_int32\_t spfMaxDelay)  
*Sets This function sets the minimum and maximum delay between a topology change, being either received in an LSA or self detected, and the SPF calculation being run.*
- `int smi_ospf_timers_spf_unset` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*Sets This function resets the SPF minimum delay and maximum delay to their default values: 5 seconds.*
- `int smi_ospf_timers_refresh_set` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int refreshInterval)  
*Sets This function sets the LSA refresh timer value.*
- `int smi_ospf_timers_refresh_unset` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*Sets This function resets the LSA refresh timer to the default value.*
- `int smi_ospf_lsa_throttle_timers_set` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t throttleTimersStartDelay, u\_int32\_t holdInterval, u\_int32\_t throttleTimersMaxDelay)  
*Sets This function sets the rate-limiting intervals for OSPF link-state advertisement (LSA) generation.*
- `int smi_ospf_lsa_throttle_timers_unset` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*Sets This function sets the rate-limiting intervals for OSPF link-state advertisement (LSA) generation to their default values.*
- `int smi_ospf_lsa_min_arrival_set` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t lsaMinArrival)  
*Sets This function sets the minimum interval to accept the same link-state advertisement (LSA) from OSPF neighbors.*
- `int smi_ospf_lsa_min_arrival_unset` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*Sets This function sets the minimum interval to accept the same link-state advertisement (LSA) from OSPF neighbors to its default value (1000 milliseconds).*
- `int smi_ospf_auto_cost_reference_bandwidth_set` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int referenceBandwidth)  
*Sets This function sets the reference bandwidth value. OSPF calculates the OSPF metric for an interface by dividing the reference bandwidth.*
- `int smi_ospf_auto_cost_reference_bandwidth_unset` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)



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

- int [smi\\_ospf\\_max\\_concurrent\\_dd\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int16\_t maxDBDescriptor)

*Sets This function sets the specified limit for the number of concurrently processed Database Descriptors.*

- int [smi\\_ospf\\_max\\_concurrent\\_dd\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*Sets This function resets the specified limit for the number of concurrently processed Database Descriptors.*

- int [smi\\_ospf\\_max\\_unuse\\_packet\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t maxUnusePackets)

*This function sets the maximum no of unused ospf packets.*

- int [smi\\_ospf\\_max\\_unuse\\_packet\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*This function sets the maximum no of unused ospf packets to default value.*

- int [smi\\_ospf\\_max\\_unuse\\_lsa\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t maxLsaPacket)

*This function sets maximum number of link state advertisement packet to specified value.*

- int [smi\\_ospf\\_max\\_unuse\\_lsa\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*Sets This function sets maximum number of link state advertisement packet to default value.*

- int [smi\\_ospf\\_overflow\\_database\\_external\\_limit\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t lsdbExternalLimit)

*Sets This function sets the maximum number of LSAs as specified.*

- int [smi\\_ospf\\_overflow\\_database\\_external\\_limit\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*This call implements the no parameter of the enable ext-ospf-multi-inst command to disable support of multiple OSPF instances on a subnet.*

- int [smi\\_ospf\\_overflow\\_database\\_external\\_interval\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int exitOverflowInterval)

*Sets This function sets the value of the time-to-recover interval of the overflow state.*

- int [smi\\_ospf\\_overflow\\_database\\_external\\_interval\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*Sets This function resets the value of the interval of the overflow state as defined by OSPF\_DEFAULT\_EXIT\_OVERFLOW\_INTERVAL.*

- int [smi\\_ospf\\_enable\\_ext\\_multi\\_inst](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId)  
*Sets This function enables multiple OSPF instances to run on a subnet.*
- int [smi\\_ospf\\_if\\_network\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, int ifNetworkType)  
*Sets This function configures the OSPF network type as specified.*
- int [smi\\_ospf\\_if\\_network\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)  
*Sets This function resets the network type to the default type.*
- int [smi\\_ospf\\_if\\_network\\_p2mp\\_nbma\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)  
*Sets This function configures an interface to Point-to-Multipoint Non-Broadcast mode.*
- int [smi\\_ospf\\_if\\_authentication\\_type\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_char ifAuthType)  
*Sets This function sets the authentication type of the current . interface.*
- int [smi\\_ospf\\_if\\_authentication\\_type\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_char ifAuthType)  
*Sets This function sets the authentication type of the interface specified by IP address.*
- int [smi\\_ospf\\_if\\_authentication\\_type\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)  
*Sets This function removes the authentication type for the current interface.*
- int [smi\\_ospf\\_if\\_authentication\\_type\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)  
*Sets This function resets the authentication type for the specified interface.*
- int [smi\\_ospf\\_if\\_priority\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_char ifPriority)  
*Sets This function sets the priority of the current interface.*
- int [smi\\_ospf\\_if\\_priority\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_char ifPriority)  
*Sets This sets the priority of the interface specified by IP address.*
- int [smi\\_ospf\\_if\\_priority\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)  
*Sets This function resets the priority of the current interface to the default value 1.*

- int [smi\\_ospf\\_if\\_priority\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)  
*Sets This function resets the priority of the interface specified. by IP address of the default value 1.*
- int [smi\\_ospf\\_if\\_mtu\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int16\_t ifMtu)  
*Sets This function sets mtu size.*
- int [smi\\_ospf\\_if\\_mtu\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)  
*Sets This function resets mtu size to default.*
- int [smi\\_ospf\\_if\\_mtu\\_ignore\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)  
*This function sets ospf not to check mtu size during database description exchange.*
- int [smi\\_ospf\\_if\\_mtu\\_ignore\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)  
*This function sets ospf not to check mtu size during database description exchange for particular address.*
- int [smi\\_ospf\\_if\\_mtu\\_ignore\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)  
*This function unconfigurs the ospf mtu ignorance of during database description exchange.*
- int [smi\\_ospf\\_if\\_mtu\\_ignore\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)  
*This function unconfigurs the ospf mtu ignorance of during database description exchange for particular address.*
- int [smi\\_ospf\\_if\\_cost\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifCost)  
*Sets This function sets the current interface output cost.*
- int [smi\\_ospf\\_if\\_cost\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifCost)  
*Sets This function sets the output cost of the interface of the specific IP address.*
- int [smi\\_ospf\\_if\\_cost\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)  
*Sets This function resets the cost for the current interface to the default value 10.*
- int [smi\\_ospf\\_if\\_cost\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)  
*Sets This function resets the cost for the specified interface to the default value 10.*

- int [smi\\_ospf\\_if\\_transmit\\_delay\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifTransmitDelay)

*This function sets the transmit delay interval (seconds) for the current interface.*

- int [smi\\_ospf\\_if\\_transmit\\_delay\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifTransmitDelay)

*This function sets the transmit delay interval (seconds) for the interface specified by IP address.*

- int [smi\\_ospf\\_if\\_transmit\\_delay\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

*Sets This function resets the transmit delay interval of the current interface.*

- int [smi\\_ospf\\_if\\_transmit\\_delay\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)

*Sets This function resets the transmit delay interval of the interface specified by IP address to the default.*

- int [smi\\_ospf\\_if\\_retransmit\\_interval\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifRetransmitInterval)

*Sets This function sets the time between LSA retransmission for adjacencies belonging to the interface.*

- int [smi\\_ospf\\_if\\_retransmit\\_interval\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifRetransmitInterval)

*Sets This function the time between LSA retransmission for adjacencies belonging to the interface by ip address.*

- int [smi\\_ospf\\_if\\_retransmit\\_interval\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

*Sets This function resets the retransmit interval of the current interface to the default.*

- int [smi\\_ospf\\_if\\_retransmit\\_interval\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)

*Sets This function resets the retransmit interval of the interface specified by IP address to the default.*

- int [smi\\_ospf\\_if\\_hello\\_interval\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifHelloInterval)

*Sets This function sets the hello interval for the current interface.*

- int [smi\\_ospf\\_if\\_hello\\_interval\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifHelloInterval)

*This function sets the hello interval for the interface specified by IP address.*

- int [smi\\_ospf\\_if\\_hello\\_interval\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)  
*This function resets the hello interval of the current interface to the default value.*
- int [smi\\_ospf\\_if\\_hello\\_interval\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, struct pal\_in4\_addr ipAddr)  
*This function resets the hello interval of the interface specified by IP address to the default value.*
- int [smi\\_ospf\\_if\\_dead\\_interval\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, u\_int32\_t ifDeadInterval)  
*This function sets the router-dead-interval for the current interface.*
- int [smi\\_ospf\\_if\\_dead\\_interval\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifDeadInterval)  
*This function sets the router-dead-interval for the interface specified by the IP address.*
- int [smi\\_ospf\\_if\\_dead\\_interval\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)  
*This function resets the dead interval of the interface specified by IP address to the default.*
- int [smi\\_ospf\\_if\\_dead\\_interval\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, struct pal\_in4\_addr ipAddr)  
*This function unsets the router-dead-interval for the interface specified by the IP address.*
- int [smi\\_ospf\\_if\\_authentication\\_key\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, char \*md5, char \*ifAuthKey)  
*This function sets the authentication key for simple password.*
- int [smi\\_ospf\\_if\\_authentication\\_key\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, struct pal\_in4\_addr ipAddr, char \*md5, char \*ifAuthKey)  
*This function sets the interface's authentication key for an area of the specified IP address with a simple password.*
- int [smi\\_ospf\\_if\\_authentication\\_key\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName)  
*This function removes the interface authentication key for an area.*
- int [smi\\_ospf\\_if\\_authentication\\_key\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrfId, char \*ifName, struct pal\_in4\_addr ipAddr)  
*This function disables the authentication key for the interface specified by IP address.*

- int [smi\\_ospf\\_if\\_message\\_digest\\_key\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_char msgDigestKeyId, char \*md5, char \*ifAuthKey)

*This function sets the MD5 key for the current interface.*

- int [smi\\_ospf\\_if\\_message\\_digest\\_key\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_char msgDigestKeyId, char \*md5, char \*ifAuthKey)

*This function sets the MD5 key for the interface specified by IP address.*

- int [smi\\_ospf\\_if\\_message\\_digest\\_key\\_get](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_char msgDigestKeyId, char \*ifAuthKey)

*This function gets the MD5 key for the mentioned interface.*

- int [smi\\_ospf\\_if\\_message\\_digest\\_key\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_char msgDigestKeyId)

*This function deletes the MD5 key for the current interface.*

- int [smi\\_ospf\\_if\\_message\\_digest\\_key\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_char msgDigestKeyId)

*This function deletes the MD5 key for the interface specified by IP address.*

- int [smi\\_ospf\\_if\\_te\\_metric\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifTEMetric)

*This function sets the TE-metric on the specified interface.*

- int [smi\\_ospf\\_if\\_te\\_metric\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

*This function unsets the TE-metric on a particular interface.*

- int [smi\\_ospf\\_if\\_database\\_filter\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

*This function suppresses all LSA during synchronization and flooding on a particular interface.*

- int [smi\\_ospf\\_if\\_database\\_filter\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)

*This function suppresses all LSA during synchronization and flooding for a particular interface by IP address.*

- int [smi\\_ospf\\_if\\_database\\_filter\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

*This function restores the forwarding of LSAs.*

- int [smi\\_ospf\\_if\\_database\\_filter\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)

*This function restores the database filter of the interface specified by IP address.*

- int [smi\\_ospf\\_if\\_disable\\_all\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

*This function disables all packet processing on a particular interface regardless whether the network area command is configured.*

- int [smi\\_ospf\\_if\\_disable\\_all\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

*This function unconfigures the ip ospf disable all command.*

- int [smi\\_ospf\\_if\\_resync\\_timeout\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifResyncTimeout)

*This function sets the timeout interval for re-synchronization. If out-of-band re-synchronization does not occur, adjacency is reset.*

- int [smi\\_ospf\\_if\\_resync\\_timeout\\_set\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifResyncTimeout)

*This function sets the time between LSA retransmission for adjacencies belonging to the interface by ip address.*

- int [smi\\_ospf\\_if\\_resync\\_timeout\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

*This function unsets the timeout interval for re-synchronization.*

- int [smi\\_ospf\\_if\\_resync\\_timeout\\_unset\\_by\\_addr](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)

*This function unsets the timeout interval for re-synchronization for particular address.*

- int [smi\\_ospf\\_vlink\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)

*This function creates a virtual interface and configures a virtual neighbor.*

- int [smi\\_ospf\\_vlink\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)

*This function destroys the specified virtual interface and deconfigures the specified virtual neighbor.*

- int [smi\\_ospf\\_vlink\\_dead\\_interval\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtDeadInterval)

*This function sets the router dead interval value for the virtual interface.*

- int [smi\\_ospf\\_vlink\\_hello\\_interval\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtHelloInterval)

*This function sets the router hello interval value for the virtual interface. The hello interval value must be the same for both ends of the virtual link.*

- int [smi\\_ospf\\_vlink\\_transmit\\_delay\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtTransmitDelay)

*This function sets the specified transmit delay (in seconds) for the virtual link.*

- int [smi\\_ospf\\_vlink\\_dead\\_interval\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)

*This function resets the router dead interval value for the virtual interface to the default value 40.*

- int [smi\\_ospf\\_vlink\\_hello\\_interval\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)

*This function resets the router hello interval value for the virtual interface to the default value 10 seconds.*

- int [smi\\_ospf\\_vlink\\_retransmit\\_interval\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)

*This function resets the retransmit interval value for the virtual interface to the default value 5 seconds.*

- int [smi\\_ospf\\_vlink\\_transmit\\_delay\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)

*This function resets the transmit delay for the interface 1 seconds.*

- int [smi\\_ospf\\_vlink\\_authentication\\_type\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtAuthType)

*This function sets the authentication type for the virtual interface.*

- int [smi\\_ospf\\_vlink\\_authentication\\_type\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)

*This function resets the authentication type for the virtual interface to NULL.*

- int [smi\\_ospf\\_vlink\\_authentication\\_key\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, char \*ifVirtAuthKey)

*This function sets the simple authentication password (type 1) for the OSPF virtual links. Simple password authentication allows a password (key) to be configured per area. To participate in the routing domain, routers in the same area must be configured with the same key.*



- int [smi\\_ospf\\_vlink\\_authentication\\_key\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)  
*This function resets the simple authentication password for the virtual link to NULL.*
- int [smi\\_ospf\\_vlink\\_message\\_digest\\_key\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, u\_char virtMsgDigestKeyId, char \*ifVirtAuthKey)  
*This function sets the MD5 authentication key for the virtual interface.*
- int [smi\\_ospf\\_vlink\\_message\\_digest\\_key\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, u\_char virtMsgDigestKeyId)  
*This function sets the MD5 authentication key for the virtual interface.*
- int [smi\\_ospf\\_summary\\_address\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask)  
*This function sets the external summary address range.*
- int [smi\\_ospf\\_summary\\_address\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask)  
*This function resets the external summary address range.*
- int [smi\\_ospf\\_summary\\_address\\_tag\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask, u\_int32\_t tag)  
*This function sets the tag value to the specified value. A tag value that can be used as a match value for controlling redistribution via route maps.*
- int [smi\\_ospf\\_summary\\_address\\_tag\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask)  
*This function resets the tag value of the external summary address range to zero.*
- int [smi\\_ospf\\_summary\\_address\\_not\\_advertise\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask)  
*This function sets the flag of the external summary address range to Not Advertise.*
- int [smi\\_ospf\\_nbr\\_static\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr)  
*This function sets the non-broadcast multi-access (nbma) neighbor.*
- int [smi\\_ospf\\_nbr\\_static\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr)

*This function deletes the static non-broadcast multi-access (nbma) neighbor.*

- int [smi\\_ospf\\_nbr\\_static\\_priority\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, u\_char neighborStaticPriority)

*This function sets the priority of the specified non-broadcast multi-access (nbma) neighbor.*

- int [smi\\_ospf\\_nbr\\_static\\_priority\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress)

*This function resets the priority of the non-broadcast multi-access (nbma) neighbor to the default value.*

- int [smi\\_ospf\\_nbr\\_static\\_poll\\_interval\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int nbrPollInterval)

*This function sets the poll interval of the non-broadcast multi-access (nbma) neighbor.*

- int [smi\\_ospf\\_nbr\\_static\\_poll\\_interval\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress)

*This function resets the poll interval of the specified non-broadcast multi-access (nbma) neighbor to the default value.*

- int [smi\\_ospf\\_nbr\\_static\\_cost\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, u\_int16\_t neighborCost)

*This function sets the cost of the specified non-broadcast multi-access (nbma) neighbor.*

- int [smi\\_ospf\\_nbr\\_static\\_cost\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress)

*This function resets the cost of the specified non-broadcast multi-access (nbma) neighbor to the default value 0.*

- int [smi\\_ospf\\_area\\_auth\\_type\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char areaAuthType)

*This function enables authentication for an area.*

- int [smi\\_ospf\\_area\\_auth\\_type\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function disables the authentication type for the area.*

- int [smi\\_ospf\\_area\\_stub\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function sets the specified area as a stub area.*

- int [smi\\_ospf\\_area\\_stub\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function removes the stub definition from the specified area.*

- int [smi\\_ospf\\_area\\_no\\_summary\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function sets the OSPF area as stub.*

- int [smi\\_ospf\\_area\\_no\\_summary\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function disables the OSPF area as stub.*

- int [smi\\_ospf\\_area\\_default\\_cost\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_int32\_t areaCost)

*This function assigns the specified cost to the default summary route used for a not so stubby area (NSSA).*

- int [smi\\_ospf\\_area\\_default\\_cost\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function resets the cost to the default value of 1.*

- int [smi\\_ospf\\_area\\_range\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, u\_char rangeMask)

*This function specifies an address range, for which to advertise a single route to other areas by the ABRs.*

- int [smi\\_ospf\\_area\\_range\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, u\_char rangeMask)

*This function deletes the area range.*

- int [smi\\_ospf\\_area\\_range\\_not\\_advertise\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, u\_char rangeMask)

*This function sets the ABR to not advertise the summary LSA for each route in a specific adjacent area.*

- int [smi\\_ospf\\_area\\_range\\_not\\_advertise\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, u\_char rangeMask)

*This function allows the area border router (ABR) to create a summary LSA for each route in specific area and advertise it in adjacent areas.*

- int [smi\\_ospf\\_area\\_range\\_substitute\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, char \*ospfAreaRange, char \*substituteAddr)

*This function summarizes routes via Matching addresses and or masks, border routers only and Announcing area range as a prefix.*

- int [smi\\_ospf\\_area\\_range\\_substitute\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, char \*ospfAreaRange)

*This function cancels the routes via Matching addresses and or masks, border routers only and Announcing area range as a prefix.*

- int [smi\\_ospf\\_area\\_filter\\_list\\_prefix\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int prefixType, char \*prefixName)

*This function sets the type of filter prefix advertised in type-3 LSAs between the OSPF areas of an ABR.*

- int [smi\\_ospf\\_area\\_filter\\_list\\_prefix\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int prefixType)

*This function cancels the filter prefix advertise.*

- int [smi\\_ospf\\_area\\_filter\\_list\\_access\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int accessType, char \*accessName)

*This function sets to filter prefixes advertise in type-3 link-state advertisement (LSAs) with the access list name between OSPF areas of an ABR.*

- int [smi\\_ospf\\_area\\_filter\\_list\\_access\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int accessType)

*This function resets the filter list access configuration to either `FILTER_IN` or `FILTER_OUT`.*

- int [smi\\_ospf\\_area\\_export\\_list\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, char \*accessListName)

*This function sets the type-3 export filter for networks announced to other areas.*

- int [smi\\_ospf\\_area\\_export\\_list\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function resets the export list.*

- int [smi\\_ospf\\_area\\_import\\_list\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, char \*accessListName)

*This function sets the import list value for the type-3 import filter.*

- int [smi\\_ospf\\_area\\_import\\_list\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function resets the import list value for the type-3 import.*

- int [smi\\_ospf\\_area\\_shortcut\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char areaShortcutType)

*This function sets the shortcut mode of the specified area. Area shortcut enables traffic to go through the nonbackbone area with a lower metric, whether or not the ABR is attached to the backbone area.*

- int [smi\\_ospf\\_area\\_shortcut\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function removes the shortcut mode of the specified area.*

- s\_int32\_t [smi\\_ospf\\_multi\\_area\\_adjacency\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, char \*ifName, struct pal\_in4\_addr mlinkNbr, int areaFormat)

*This call implements the area multi-area-adjacency command to enable the multiple-area adjacency on specified interface for the given area ID.*

- s\_int32\_t [smi\\_ospf\\_multi\\_area\\_adjacency\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char \*ifName, struct pal\_in4\_addr mlinkNbr)

*This call implements the no parameter of the area multi-area-adjacency command to disable multi-areaadjacency.*

- int [smi\\_ospf\\_set\\_nssa\\_stability\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int nssaStabilityInterval, u\_int32\_t vrId)

*This function sets the number of seconds an elected translator should continue to perform its translation duties after it has determined its services are no longer required.*

- int [smi\\_ospf\\_area\\_nssa\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function defines an area as Not So Stubby Area.*

- int [smi\\_ospf\\_area\\_nssa\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function removes the NSSA designation from the specified area.*

- int [smi\\_ospf\\_area\\_nssa\\_translator\\_role\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char nssaTranslatorRole)

*This function sets the Area border router to be the translator between the types. This is applied only if the area type is NSSA.*

- int [smi\\_ospf\\_area\\_nssa\\_translator\\_role\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*Removes the value of the translator parameter of the area.*

- int [smi\\_ospf\\_area\\_nssa\\_no\\_redistribution\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function sets OSPF redistribution in not allowed to the stub.*

- int [smi\\_ospf\\_area\\_nssa\\_no\\_redistribution\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function resets OSPF: redistribution in not allowed to the stub.*

- int [smi\\_ospf\\_area\\_nssa\\_default\\_originate\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)  
*Sets the originate Type-7 default LSA into the NSSA.*
- int [smi\\_ospf\\_area\\_nssa\\_default\\_originate\\_metric\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, int nssaDefaultOriginMetric)  
*Sets the default metric for the routes originated from this NSSA router.*
- int [smi\\_ospf\\_area\\_nssa\\_default\\_originate\\_metric\\_type\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, int nssaDefaultOriginMetricType)  
*Sets the default metric type for the routes originated from this NSSA router.*
- int [smi\\_ospf\\_area\\_nssa\\_default\\_originate\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)  
*Unsets the originate Type-7 default LSA into the NSSA.*
- int [smi\\_ospf\\_area\\_nssa\\_stability\\_interval\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_int32\_t nssaStabilityInterval)  
*This function sets the NSSA stability interval.*
- int [smi\\_ospf\\_opaque\\_link\\_lsa\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr ipAddr, u\_char opaqueType, u\_int32\_t opaqueId, char \*opaqueData, u\_int32\_t opaqueLen)  
*This function generates the specified AS-wide Opaque LSA.*
- int [smi\\_ospf\\_opaque\\_area\\_lsa\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char opaqueType, u\_int32\_t opaqueId, char \*opaqueData, u\_int32\_t opaqueLen)  
*This function generates area Opaque LSAs.*
- int [smi\\_ospf\\_opaque\\_as\\_lsa\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_char opaqueType, u\_int32\_t opaqueId, char \*opaqueData, u\_int32\_t opaqueLen)  
*This function generates Autonomous System (AS) area opaque LSAs.*
- int [smi\\_ospf\\_capability\\_opaque\\_lsa\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*This function sets the opaque capability for an OSPF process.*
- int [smi\\_ospf\\_capability\\_opaque\\_lsa\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*This function unsets the opaque capability for an OSPF process.*
- int [smi\\_ospf\\_capability\\_traffic\\_engineering\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*This function sets the traffic engineering capability for an OSPF process.*

- int [smi\\_ospf\\_capability\\_traffic\\_engineering\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*This function resets the traffic engineering capability for an OSPF process.*

- int [smi\\_ospf\\_te\\_link\\_flood\\_scope\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName, int ospfProcessId, struct pal\_in4\_addr areaId, int areaFormat)

*This function sets the flooding scope (area and OSPF instance) of a specified GMPLS TE link. The TE link is flooded over control links in the specified area and in the OSPF instance.*

- int [smi\\_ospf\\_te\\_link\\_flood\\_scope\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName, int ospfProcessId, struct pal\_in4\_addr areaId)

*This function unsets the flooding scope of a specified GMPLS TE link. If the TE link is advertised in an area, this function withdraws TE link from that area.*

- int [smi\\_ospf\\_telink\\_te\\_metric\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName, u\_int32\_t teMetric)

*This function sets the traffic engineering metric for a GMPLS TE-link.*

- int [smi\\_ospf\\_telink\\_te\\_metric\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName)

*This function sets the traffic engineering metric for a GMPLS TE link to the default value.*

- int [smi\\_ospf\\_opaque\\_te\\_link\\_local\\_lsa\\_enable](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName)

*This function enables the exchange of TE link local LSA for a specified GMPLS TE link and determines the remote interface ID of the TE link.*

- int [smi\\_ospf\\_opaque\\_te\\_link\\_local\\_lsa\\_disable](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName)

*This function disables the exchange of TE link local LSA for the specified TE link.*

- int [smi\\_ospf\\_capability\\_cspf\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*This function sets the Constrained Shortest Path First (CSPF) capability for an OSPF process. The CSPF protocol module relies on the OSPF database to calculate the shortest path through the network.*

- int [smi\\_ospf\\_capability\\_cspf\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*This function disables the CSPF capability for an OSPF process.*

- int [smi\\_ospf\\_cspf\\_better\\_protection\\_type](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, bool\_t cspfProtectionType)

*This function enables/disables the default cspf protection type.*

- int [smi\\_ospf\\_enable\\_db\\_summary\\_opt](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*This function enables the OSPF Database Summary List optimization. The OSPF\_DB\_SUMMARY\_OPT flag is set to indicate the feature is enabled.*

- int [smi\\_ospf\\_disable\\_db\\_summary\\_opt](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

*This function disables the OSPF Database Summary List optimization. The OSPF\_DB\_SUMMARY\_OPT flag is unset to indicate the feature is disabled.*

- int [smi\\_ospf\\_redistribute\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int secondaryOspfProcessId, int routeSourceType, int redistMetricType, int redistMetricValue)

*Sets OSPF to redistribute routes learned from different sources into another OSPF instance.*

- int [smi\\_ospf\\_redistribute\\_default\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceOrigin, int redistMetricType, int redistMetricValue)

*Sets OSPF to redistribute default routes into another OSPF instance.*

- int [smi\\_ospf\\_redist\\_proto\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)

*Sets OSPF to redistribute connected/kernel/static routes and routes from different routing protocols into another OSPF instance.*

- int [smi\\_ospf\\_redist\\_proto\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)

*Sets OSPF not to redistribute connected/kernel/static routes and routes from different routing protocols into another OSPF instance.*

- int [smi\\_ospf\\_redist\\_metric\\_type\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int redistMetricType, int secondaryOspfProcessId)

*Sets OSPF to redistribute routes with an external metric type.*

- int [smi\\_ospf\\_redist\\_metric\\_type\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)

*Sets OSPF not to redistribute routes with an external metric type.*

- int [smi\\_ospf\\_redist\\_metric\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int redistMetricValue, int secondaryOspfProcessId)

*Sets OSPF to redistribute routes with an external metric value.*



- int [smi\\_ospf\\_redist\\_metric\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)  
*Sets OSPF not to redistribute routes with an external metric value.*
- int [smi\\_ospf\\_redist\\_tag\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, u\_int32\_t redistTag, int secondaryOspfProcessId)  
*Sets OSPF to redistribute routes with a tag value to be used as a match for controlling redistribution via route maps.*
- int [smi\\_ospf\\_redist\\_tag\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)  
*Sets OSPF not to redistribute routes with a tag value that was set to use as a match for controlling redistribution via route maps.*
- int [smi\\_ospf\\_redist\\_default\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceOrigin)  
*Sets OSPF to redistribute default route into an OSPF instance.  
When set, the router acts like an ASBR to redistribute routes into an OSPF instance.  
By default an ASBR does not generate a default route into an OSPF instance.*
- int [smi\\_ospf\\_redist\\_default\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*Sets OSPF not to redistribute default route into an OSPF instance.*
- int [smi\\_ospf\\_distribute\\_list\\_out\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId, char \*accessListName)  
*Sets OSPF to distribute routing updates from particular routing protocol into another OSPF instance (i.e filters networks from particular outgoing routing updates).*
- int [smi\\_ospf\\_distribute\\_list\\_out\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId, char \*accessListName)  
*Unsets OSPF from distributing routing updates from particular routing protocol into another OSPF instance (i.e filters networks from particular outgoing routing updates).*
- int [smi\\_ospf\\_distribute\\_list\\_in\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*accessListName)  
*Sets OSPF to allow routing updates from particular access list into this OSPF instance (i.e filters networks from particular access list at ingress).*
- int [smi\\_ospf\\_distribute\\_list\\_in\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*accessListName)  
*Unsets OSPF to allow routing updates from particular access list into this OSPF instance (i.e filters networks from particular access list at ingress).*

- `int smi_ospf_routermap_set` (struct `smiclient_globals *azg`, `u_int32_t` `vrId`, `int` `ospfProcessId`, `int` `routeSourceType`, `char *routeMapName`, `int` `secondaryOspfProcessId`)  
*Sets OSPF to redistribute routes via an route-map reference.  
When set, OSPF does not look for default network to redistribute routes.*
- `int smi_ospf_routermap_unset` (struct `smiclient_globals *azg`, `u_int32_t` `vrId`, `int` `ospfProcessId`, `int` `routeSourceType`, `int` `secondaryOspfProcessId`)  
*Sets OSPF not to redistribute routes via an route-map reference.  
When unset, OSPF look for default network to redistribute routes.*
- `int smi_ospf_routermap_default_set` (struct `smiclient_globals *azg`, `u_int32_t` `vrId`, `int` `ospfProcessId`, `char *routeMapName`)  
*Sets the default route-map reference for OSPF to redistribute routes via.*
- `int smi_ospf_routermap_default_unset` (struct `smiclient_globals *azg`, `u_int32_t` `vrId`, `int` `ospfProcessId`)  
*Unsets the default route-map reference for OSPF to redistribute routes via.*
- `int smi_ospf_default_metric_set` (struct `smiclient_globals *azg`, `u_int32_t` `vrId`, `int` `ospfProcessId`, `int` `metricValue`)  
*Sets the default metric value for OSPF to redistribute routes.  
A default metric facilitates redistributing routes with incompatible metrics.*
- `int smi_ospf_default_metric_unset` (struct `smiclient_globals *azg`, `u_int32_t` `vrId`, `int` `ospfProcessId`)  
*Unsets the default metric value for OSPF to redistribute routes.*
- `int smi_ospf_distance_all_set` (struct `smiclient_globals *azg`, `u_int32_t` `vrId`, `int` `ospfProcessId`, `int` `adminDistance`)  
*Sets the OSPF administrative distance.  
The administrative distance rates the trustworthiness of a routing information source.  
The higher distance value implies the lower trustworthiness.*
- `int smi_ospf_distance_all_unset` (struct `smiclient_globals *azg`, `u_int32_t` `vrId`, `int` `ospfProcessId`, `int` `adminDistance`)  
*Unsets the OSPF administrative distance.*
- `int smi_ospf_distance_intra_area_set` (struct `smiclient_globals *azg`, `u_int32_t` `vrId`, `int` `ospfProcessId`, `int` `intraAreaDistance`)  
*Sets the OSPF administrative distance for all routes within an area (i.e., intra-area).  
The administrative distance rates the trustworthiness of a routing information source.  
The higher distance value implies the lower trustworthiness.*
- `int smi_ospf_distance_intra_area_unset` (struct `smiclient_globals *azg`, `u_int32_t` `vrId`, `int` `ospfProcessId`)  
*Unsets the OSPF administrative distance for all routes within an area (i.e., intra-area).*

- int [smi\\_ospf\\_distance\\_inter\\_area\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int interAreaDistance)  
*Sets the OSPF administrative distance for all routes from one area to another area (i.e., inter-area).*  
*The administrative distance rates the trustworthiness of a routing information source.*  
*The higher distance value implies the lower trustworthiness.*
- int [smi\\_ospf\\_distance\\_inter\\_area\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*Unsets the OSPF administrative distance for all routes from one area to another area (i.e., inter-area).*
- int [smi\\_ospf\\_distance\\_external\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int externalDistance)  
*Sets the OSPF administrative distance for all routes learned from other routing domains learned via redistribution.*  
*The administrative distance rates the trustworthiness of a routing information source.*  
*The higher distance value implies the lower trustworthiness.*
- int [smi\\_ospf\\_distance\\_external\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*Unsets the OSPF administrative distance for all routes learned from other routing domains learned via redistribution.*
- int [smi\\_ospf\\_distance\\_source\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t adminDistance, struct pal\_in4\_addr sourceIpAddr, u\_char masklen, char \*accessListName)  
*Sets the OSPF administrative distance to prefixes whose nexthop matches the given source IP address.*  
*The administrative distance rates the trustworthiness of a routing information source.*  
*The higher distance value implies the lower trustworthiness.*
- int [smi\\_ospf\\_distance\\_source\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr sourceIpAddr, u\_char masklen, char \*accessListName)  
*Unsets the OSPF administrative distance to prefixes whose nexthop matches the given source IP address.*
- int [smi\\_ospf\\_capability\\_restart\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int restartMethod)  
*Enables the OSPF restart capability by graceful restart or signalling restart method.*
- int [smi\\_ospf\\_capability\\_restart\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*Disables the OSPF restart capability.*
- int [smi\\_ospf\\_restart\\_helper\\_policy\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int restartHelperPolicy)

*Sets the helper behavior for OSPF graceful restart.*

- int [smi\\_ospf\\_restart\\_helper\\_policy\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId)

*Resets the helper behavior to default (i.e., always accept).*

- int [smi\\_ospf\\_restart\\_helper\\_never\\_router\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, struct pal\_in4\_addr nbrRouterId)

*Sets the particular neighbor ID to never act as helper.*

- int [smi\\_ospf\\_restart\\_helper\\_never\\_router\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, struct pal\_in4\_addr nbrRouterId)

*Unsets the particular neighbor ID that was to never act as helper.*

- int [smi\\_ospf\\_restart\\_helper\\_never\\_router\\_unset\\_all](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId)

*Removes all neighbor IDs from the router helper list.*

- int [smi\\_ospf\\_restart\\_helper\\_grace\\_period\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int restartHelperPeriod)

*Sets the maximum grace period allowed to be as helper router for restarting a router.*

- int [smi\\_ospf\\_restart\\_helper\\_grace\\_period\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId)

*Removes the maximum grace period allowed that was set to be as helper router for restarting a router.*

- int [smi\\_ospf\\_get\\_router\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr \*routerId, u\_int32\_t vrId)

*Gets the 32-bit integer ID that uniquely identifies the router in the Autonomous System.*

- int [smi\\_ospf\\_get\\_admin\\_stat](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*adminStatus, u\_int32\_t vrId)

*Gets the administrative status of given OSPF instance in the router.*

*The value 'enabled' denotes that the OSPF Process is active on at least one interface; 'disabled' disables it on all interfaces.*

- int [smi\\_ospf\\_get\\_version\\_number](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*ospfVersion, u\_int32\_t vrId)

*Gets the OSPF version number.*

- int [smi\\_ospf\\_get\\_area\\_bdr\\_rtr\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*areaBdrRouterStatus, u\_int32\_t vrId)

*Gets a flag to note whether this router is an Area Border Router.*

- int [smi\\_ospf\\_get\\_asbdr\\_rtr\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*areaASBDRRouterStatus, u\_int32\_t vrId)

*Gets a flag to note whether this router is configured as an Autonomous System Border Router.*

- int [smi\\_ospf\\_set\\_asbdr\\_rtr\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, int areaASBDRRouterStatus, u\_int32\_t vrId)

*Sets a flag to note whether this router is configured as an Autonomous System Border Router.*

- int [smi\\_ospf\\_get\\_extern\\_lsa\\_count](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*lsaCount, u\_int32\_t vrId)

*Gets the number of external (LS type-5) link state advertisements in the link state database.*

- int [smi\\_ospf\\_get\\_extern\\_lsa\\_cksum\\_sum](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*lsaChecksumSum, u\_int32\_t vrId)

*Gets the 32-bit sum of the LS checksums of the external link state advertisements contained in the link state database.*

*This sum can be used to determine if there has been a change in a router's link state database and to compare the link state database of two routers.*

*The value should be treated as unsigned when comparing two sums of checksums.*

- int [smi\\_ospf\\_get\\_tos\\_support](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*TOSStatus, u\_int32\_t vrId)

*Gets a flag to note whether router has support for type-of-service routing.*

- int [smi\\_ospf\\_get\\_originate\\_new\\_lsas](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*lsaCount, u\_int32\_t vrId)

*Gets the number of new link state advertisements that have been originated from this router.*

*This number is incremented each time the router originates a new LSA.*

- int [smi\\_ospf\\_get\\_rx\\_new\\_lsas](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*lsaCount, u\_int32\_t vrId)

*Gets the number of link state advertisements received that are determined to be new instantiations.*

*This number does not include newer instantiations of self-originated link state advertisements.*

- int [smi\\_ospf\\_get\\_ext\\_lsdb\\_limit](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*lsdbLimitCount, u\_int32\_t vrId)

*Gets the maximum number of non-default AS-external LSAs entries that can be stored in the link state database.*

*If the value is -1, then there is no limit.*

*When the number of non-default AS-external LSAs in a router's link state database reaches ospfExtLsdbLimit, the router enters overflow state.*

*The router never holds more than ospfExtLsdbLimit non-default AS-external LSAs in its database.*

*OspfExtLsdbLimit MUST be set identically in all routers attached to the OSPF backbone and/or any regular OSPF area (i.e., OSPF stub areas and NSSAs are excluded).*

- `int smi_ospf_set_multicast_extensions` (struct smiclient\_globals \*azg, int ospf-ProcessId, int multicastExtStatus, u\_int32\_t vrId)

*Sets a bit mask indicating whether the router is forwarding IP multicast (Class D) datagrams based on the algorithms defined in the multicast extensions to OSPF.*

*Bit 0, if set, indicates that the router can forward IP multicast datagrams in the router's directly attached areas (called intra-area multicast routing).*

*Bit 1, if set, indicates that the router can forward IP multicast datagrams between OSPF areas (called inter-area multicast routing).*

*Bit 2, if set, indicates that the router can forward IP multicast datagrams between Autonomous Systems (called inter-AS multicast routing).*

*Only certain combinations of bit settings are allowed, namely: 0 (no multicast forwarding is enabled), 1 (intra-area multicasting only), 3 (intra-area and inter-area multicasting), 5 (intra-area and inter-AS multicasting), and 7 (multicasting everywhere). By default, no multicast forwarding is enabled.*

- `int smi_ospf_get_exit_overflow_interval` (struct smiclient\_globals \*azg, int ospf-ProcessId, int \*exitOvrflwInterval, u\_int32\_t vrId)

*Gets the number of seconds that, after entering OverflowState, a router will attempt to leave OverflowState. This allows the router to again originate non-default AS-external LSAs.*

*When set to 0, the router will not leave overflow state until restarted.*

- `int smi_ospf_get_demand_extensions` (struct smiclient\_globals \*azg, int ospf-ProcessId, int \*demandExtStatus, u\_int32\_t vrId)

*Gets the flag to note whether this router supports demand routing.*

- `int smi_ospf_get_area_id` (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr \*retAreaId, u\_int32\_t vrId)

*Gets the 32-bit integer uniquely identifying an area. Area ID 0.0.0.0 is used for the OSPF backbone.*

- `int smi_ospf_get_auth_type` (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*areaAuthType, u\_int32\_t vrId)

*Gets the authentication type specified for an area.*

- `int smi_ospf_get_import_as_extern` (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*areaType, u\_int32\_t vrId)

*Gets the OSPF area type that indicates if an area is a stub area, NSSA, or standard area. Type-5 AS-external LSAs and type-11 Opaque LSAs are not imported into stub areas or NSSAs. NSSAs import AS-external data as type-7 LSAs.*

- `int smi_ospf_set_import_as_extern` (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, int areaType)

*Sets the OSPF area type that indicates if an area is a stub area, NSSA, or standard area. Type-5 AS-external LSAs and type-11 Opaque LSAs are not imported into stub areas or NSSAs. NSSAs import AS-external data as type-7 LSAs.*

- `int smi_ospf_get_spf_runs` (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*spfCount, u\_int32\_t vrId)

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

- int [smi\\_ospf\\_get\\_area\\_bdr\\_rtr\\_count](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*areaBdrRouterCount, u\_int32\_t vrId)

*Gets the 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\\_ospf\\_get\\_asbdr\\_rtr\\_count](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*areaASBDRRouterCount, u\_int32\_t vrId)

*Gets the total number of Autonomous System Border Routers reachable within this area. This is initially zero and is calculated in each Shortest Path First (SPF) pass.*

- int [smi\\_ospf\\_get\\_area\\_lsa\\_count](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*areaLsaCount, u\_int32\_t vrId)

*Gets the total number of link state advertisements in this area's link state database, excluding AS-external LSAs.*

- int [smi\\_ospf\\_get\\_area\\_lsa\\_cksum\\_sum](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*areaLsaChecksumSum, u\_int32\_t vrId)

*Gets 32-bit sum of the link state advertisements' LS checksums contained in this area's link state database. This sum excludes external (LS type-5) link state advertisements. The sum can be used to determine if there has been a change in a router's link state database, and to compare the link state database of two routers. The value should be treated as unsigned when comparing two sums of checksums.*

- int [smi\\_ospf\\_get\\_area\\_summary](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*areaSummary, u\_int32\_t vrId)

*Gets a flag that controls the import of summary LSAs into stub and NSSA areas. It has no effect on other areas. If it is noAreaSummary, the router will not originate summary LSAs into the stub or NSSA area. It will rely entirely on its default route. If it is sendAreaSummary, the router will both summarize and propagate summary LSAs.*

- int [smi\\_ospf\\_get\\_area\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*areaStatus, u\_int32\_t vrId)

*Gets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_set\\_area\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int areaStatus, u\_int32\_t vrId)

*Sets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_get\\_area\\_nssa\\_translator\\_role](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*nssaTranslatorRole, u\_int32\_t vrId)

*Gets an object that indicates an NSSA border router's ability to perform NSSA translation of type-7 LSAs into type-5 LSAs.*

- int [smi\\_ospf\\_get\\_area\\_nssa\\_translator\\_state](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*nssaTranslatorState, u\_int32\_t vrId)

*Gets an object that indicates if and how an NSSA border router is performing NSSA translation of type-7 LSAs into type-5 LSAs. When this object is set to enabled, the NSSA Border router's OspfAreaNssaExtTranslatorRole has been set to always. When this object is set to elected, a candidate NSSA Border router is Translating type-7 LSAs into type-5. When this object is set to disabled, a candidate NSSA border router is NOT translating type-7 LSAs into type-5.*

- int [smi\\_ospf\\_get\\_area\\_nssa\\_translator\\_stability\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*nssaStabilityInterval, u\_int32\_t vrId)

*Gets 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\\_ospf\\_get\\_area\\_nssa\\_translator\\_events](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*nssaTranlatorChangeCnt, u\_int32\_t vrId)

*Gets an object that indicates the number of translator state changes that have occurred since the last boot-up.*

- int [smi\\_ospf\\_get\\_stub\\_area\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int typeOfService, struct pal\_in4\_addr \*stubAreaId, u\_int32\_t vrId)

*Gets 32-bit identifier for the stub area. On creation, this can be derived from the instance.*

- int [smi\\_ospf\\_get\\_stub\\_tos](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int typeOfService, int \*stubTOS, u\_int32\_t vrId)

*Gets the Type of Service associated with the metric. On creation, this can be derived from the instance.*

- int [smi\\_ospf\\_get\\_stub\\_metric](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int typeOfService, int \*stubMetric, u\_int32\_t vrId)

*Gets the metric value applied at the indicated Type of Service. By default, this equals the least metric at the Type of Service among the interfaces to other areas.*

- int [smi\\_ospf\\_get\\_stub\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int typeOfService, int \*stubStatus, u\_int32\_t vrId)

*Gets the object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_get\\_stub\\_metric\\_type](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int tos, int \*stubMetricType, u\_int32\_t vrId)

*Gets the object that displays the type of metric advertised as a default route.*



- int [smi\\_ospf\\_get\\_lsdb\\_area\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int lsaType, struct pal\_in4\_addr lsId, struct pal\_in4\_addr routerId, struct pal\_in4\_addr \*lsdbAreaId, u\_int32\_t vrId)

*Gets the 32-bit identifier of the area from which the LSA was received.*

- int [smi\\_ospf\\_get\\_lsdb\\_type](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int lsaType, struct pal\_in4\_addr lsId, struct pal\_in4\_addr routerId, int \*lsdbType, u\_int32\_t vrId)

*Gets the type of the link state advertisement. Each link state type has a separate advertisement format. Note: External link state advertisements are permitted for backward compatibility, but should be displayed in the ospfAsLsdbTable rather than here.*

- int [smi\\_ospf\\_get\\_lsdb\\_lsId](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int lsaType, struct pal\_in4\_addr lsId, struct pal\_in4\_addr routerId, struct pal\_in4\_addr \*lsdbLsId, u\_int32\_t vrId)

*Gets the Link State ID is an LS Type Specific field containing either a Router ID or an IP address; it identifies the piece of the routing domain that is being described by the advertisement.*

- int [smi\\_ospf\\_get\\_lsdb\\_router\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int lsaType, struct pal\_in4\_addr lsId, struct pal\_in4\_addr routerId, struct pal\_in4\_addr \*lsdbRouterId, u\_int32\_t vrId)

*Gets the 32-bit number that uniquely identifies the originating router in the Autonomous System.*

- int [smi\\_ospf\\_get\\_lsdb\\_sequence](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int lsaType, struct pal\_in4\_addr lsId, struct pal\_in4\_addr routerId, int \*lsdbSequence, u\_int32\_t vrId)

*Gets the sequence number field is a signed 32-bit integer. It starts with the value '80000001'h, or -'7FFFFFFF'h, and increments until '7FFFFFFF'h. Thus, a typical sequence number will be very negative. It is used to detect old and duplicate Link State Advertisements. The space of sequence numbers is linearly ordered. The larger the sequence number, the more recent the advertisement.*

- int [smi\\_ospf\\_get\\_lsdb\\_age](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int lsaType, struct pal\_in4\_addr lsId, struct pal\_in4\_addr routerId, int \*lsdbAge, u\_int32\_t vrId)

*Gets the age of the link state advertisement in seconds.*

- int [smi\\_ospf\\_get\\_lsdb\\_checksum](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int lsaType, struct pal\_in4\_addr lsId, struct pal\_in4\_addr routerId, int \*lsdbChecksum, u\_int32\_t vrId)

*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. The checksum used is the same that is used for ISO connectionless datagrams; it is commonly referred to as the Fletcher checksum.*

- int [smi\\_ospf\\_get\\_lsdb\\_advertisement](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int lsaType, struct pal\_in4\_addr lsId, struct pal\_in4\_addr routerId, struct smi\_lsa\_header \*lsa, size\_t \*lsdbAdv, u\_int32\_t vrId)

*Gets the entire link state advertisement, including its header. Note that for variable length LSAs, SNMP agents may not be able to return the largest string size.*

- int [smi\\_ospf\\_get\\_area\\_range\\_area\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, struct pal\_in4\_addr \*areaRangeAreaId, u\_int32\_t vrId)

*Gets the area that the address range is to be found within.*

- int [smi\\_ospf\\_get\\_area\\_range\\_net](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, struct pal\_in4\_addr \*areaRangeNet, u\_int32\_t vrId)

*Gets the IP address of the net or subnet indicated by the range.*

- int [smi\\_ospf\\_get\\_area\\_range\\_mask](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, struct pal\_in4\_addr \*areaRangeMask, u\_int32\_t vrId)

*Gets the subnet mask that pertains to the net or subnet.*

- int [smi\\_ospf\\_get\\_area\\_range\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, int \*areaRangeStatus, u\_int32\_t vrId)

*Gets the object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_get\\_area\\_range\\_effect](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, int \*areaRangeEffect, u\_int32\_t vrId)

*Gets the object that permits management of subnets subsumed by ranges either trigger the advertisement of the indicated summary (advertiseMatching) or result in the subnet's not being advertised at all outside the area.*

- int [smi\\_ospf\\_get\\_host\\_ip\\_address](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int typeOfService, struct pal\_in4\_addr \*hostIpAddr, u\_int32\_t vrId)

*Gets the IP address of the host.*

- int [smi\\_ospf\\_get\\_host\\_tos](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr hostAddr, int typeOfService, int \*hostTypeOfService, u\_int32\_t vrId)

*Gets the Type of Service of the route being configured.*

- int [smi\\_ospf\\_get\\_host\\_metric](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr hostAddr, int typeOfService, int \*hostMetric, u\_int32\_t vrId)

*Gets the metric to be advertised.*

- int [smi\\_ospf\\_get\\_host\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr hostAddr, int typeOfService, int \*hostStatus, u\_int32\_t vrId)

*Gets the object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_get\\_host\\_area\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr hostAddr, int typeOfService, struct pal\_in4\_addr \*hostAreaId, u\_int32\_t vrId)

*Gets the OSPF area to which the host belongs. Deprecated by ospfHostCfgAreaID. facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_get\\_if\\_ip\\_address](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, struct pal\_in4\_addr \*ifIpAddr, u\_int32\_t vrId)

*Gets the IP address of this OSPF interface.*

- int [smi\\_ospf\\_get\\_address\\_less\\_if](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr addr, int ifIndex, int \*addrLessIf, u\_int32\_t vrId)

*Gets the ifIndex for the purpose of easing the instancing of addressed and addressless interfaces; this variable takes the value 0 on interfaces with IP addresses and the corresponding value of ifIndex for interfaces having no IP address.*

- int [smi\\_ospf\\_get\\_if\\_area\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, struct pal\_in4\_addr \*areaId, u\_int32\_t vrId)

*Gets the 32-bit integer uniquely identifying the area to which the interface connects. Area ID 0.0.0.0 is used for the OSPF backbone.*

- int [smi\\_ospf\\_get\\_if\\_type](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifType, u\_int32\_t vrId)

*Gets the OSPF interface type. By way of a default, this field may be intuited from the corresponding value of ifType. Broadcast LANs, such as Ethernet and IEEE 802.5, take the value 'broadcast', X.25 and similar technologies take the value 'nbma', and links that are definitively point to point take the value 'pointToPoint'.*

- int [smi\\_ospf\\_get\\_if\\_admin\\_stat](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifAdminStat, u\_int32\_t vrId)

*Gets the administrative status of the OSPF interface. The value formed on the interface, and the interface will be advertised as an internal route to some area. The value 'disabled' denotes that the interface is external to OSPF.*

- int [smi\\_ospf\\_get\\_if\\_rtr\\_priority](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifRouterPriority, u\_int32\_t vrId)

*Gets the priority of this interface. Used in multi-access networks, this field is used in the designated router election algorithm. The value 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, routers will use their Router ID as a tie breaker.*

- int [smi\\_ospf\\_get\\_if\\_transit\\_delay](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr addr, int ifIndex, int \*ifTransmitDelay, u\_int32\_t vrid)

*Gets the transit-delay value of the OSPF interface, which is an estimate of the number of seconds required to transmit a link-state update packet through this interface. Note that the minimal value SHOULD be 1 second.*

- int [smi\\_ospf\\_get\\_if\\_retrans\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifRetransmitInterval, u\_int32\_t vrid)

*Gets the interval, in seconds, between link-state-advertisement retransmissions, for adjacencies belonging to this interface. This value is also used when retransmitting database description and Link State request packets. Note that minimal value SHOULD be 1 second.*

- int [smi\\_ospf\\_get\\_if\\_hello\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifHelloInterval, u\_int32\_t vrid)

*Gets the interval, in seconds, between the Hello packets that the router sends on the interface. This value must be the same for all routers attached to a common network.*

- int [smi\\_ospf\\_get\\_if\\_rtr\\_dead\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifDeadInterval, u\_int32\_t vrid)

*Gets the dead interval, in seconds, that a router's Hello packets have not been seen before its neighbors declare the router down. This should be some multiple of the Hello interval. This value must be the same for all routers attached to a common network.*

- int [smi\\_ospf\\_get\\_if\\_poll\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifPollInterval, u\_int32\_t vrid)

*Gets the interval, in seconds, between the Hello packets sent to an inactive non-broadcast multi-access neighbor.*

- int [smi\\_ospf\\_get\\_if\\_state](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifstate, u\_int32\_t vrid)

*Gets the OSPF interface state.*

- int [smi\\_ospf\\_get\\_if\\_designated\\_router](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, struct pal\_in4\_addr \*ifDesigRouter, u\_int32\_t vrid)

*Gets the IP address of the designated router.*

- int [smi\\_ospf\\_get\\_if\\_backup\\_designated\\_router](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, struct pal\_in4\_addr \*ifBkpDesigRouter, u\_int32\_t vrid)

*Gets the IP address of the back-up designated router.*

- int [smi\\_ospf\\_get\\_if\\_events](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifEvents, u\_int32\_t vrId)  
*Gets the of times the interface has changed state and an error has occurred.*
- int [smi\\_ospf\\_get\\_if\\_auth\\_key](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, char \*ifAuthKey, u\_int32\_t vrId)  
*Gets the OSPF authentication key.*
- int [smi\\_ospf\\_get\\_if\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifStatus, u\_int32\_t vrId)  
*Gets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*
- int [smi\\_ospf\\_get\\_if\\_multicast\\_forwarding](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifMulticastForward, u\_int32\_t vrId)  
*Gets the way multicasts should be forwarded on this interface: not forwarded, forwarded as data link multicasts, or forwarded as data link unicasts. Data link multicasting is not meaningful on point-to-point and nbma interfaces, and setting ospf-MulticastForwarding to 0 effectively disables all multicast forwarding.*
- int [smi\\_ospf\\_get\\_if\\_demand](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifDemand, u\_int32\_t vrId)  
*Gets the variable that indicates whether Demand OSPF procedures (hello suppression to FULL neighbors and setting the DoNotAge flag on propagated LSAs) should be performed on the interface.*
- int [smi\\_ospf\\_get\\_if\\_auth\\_type](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int \*ifAuthType, u\_int32\_t vrId)  
*Gets the authentication type specified for an interface.*
- int [smi\\_ospf\\_set\\_if\\_area\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, struct pal\_in4\_addr areaId, u\_int32\_t vrId)  
*Sets the 32-bit integer uniquely identifying the area to which the interface connects. Area ID 0.0.0.0 is used for the OSPF backbone.*
- int [smi\\_ospf\\_set\\_if\\_admin\\_stat](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifAdminStat, u\_int32\_t vrId)  
*Sets the administrative status of the OSPF interface. The value formed on the interface, and the interface will be advertised as an internal route to some area. The value 'disabled' denotes that the interface is external to OSPF.*
- int [smi\\_ospf\\_set\\_if\\_rtr\\_priority](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifRouterPriority, u\_int32\_t vrId)

*Sets the priority of this interface. Used in multi-access networks, this field is used in the designated router election algorithm. The value 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, routers will use their Router ID as a tie breaker.*

- int [smi\\_ospf\\_set\\_if\\_transit\\_delay](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifTransmitDelay, u\_int32\_t vrId)

*Sets the transit-delay value of the OSPF interface, which is an estimate of the number of seconds required to transmit a link-state update packet through this interface. Note that the minimal value SHOULD be 1 second.*

- int [smi\\_ospf\\_set\\_if\\_retrans\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifRetransmitInterval, u\_int32\_t vrId)

*Sets the interval, in seconds, between link-state-advertisement retransmissions, for adjacencies belonging to this interface. This value is also used when retransmitting database description and Link State request packets. Note that minimal value SHOULD be 1 second.*

- int [smi\\_ospf\\_set\\_if\\_hello\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifHelloInterval, u\_int32\_t vrId)

*Sets the interval, in seconds, between the Hello packets that the router sends on the interface. This value must be the same for all routers attached to a common network.*

- int [smi\\_ospf\\_set\\_if\\_rtr\\_dead\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifDeadInterval, u\_int32\_t vrId)

*Sets the dead interval, in seconds, that a router's Hello packets have not been seen before its neighbors declare the router down. This should be some multiple of the Hello interval. This value must be the same for all routers attached to a common network.*

- int [smi\\_ospf\\_set\\_if\\_poll\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifPollInterval, u\_int32\_t vrId)

*Sets the interval, in seconds, between the Hello packets sent to an inactive non-broadcast multi-access neighbor.*

- int [smi\\_ospf\\_set\\_if\\_auth\\_key](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int authKeyLength, char \*ifAuthKey, u\_int32\_t vrId)

*Sets the OSPF authentication key.*

- int [smi\\_ospf\\_set\\_if\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifStatus, u\_int32\_t vrId)

*Sets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_set\\_if\\_auth\\_type](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifAuthType, u\_int32\_t vrId)  
*Sets the authentication type specified for an interface.*
- int [smi\\_ospf\\_get\\_if\\_metric\\_ip\\_address](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int typeOfService, struct pal\_in4\_addr \*ifMetricIpAddr, u\_int32\_t vrId)  
*Gets the IP address of this OSPF interface.*
- int [smi\\_ospf\\_get\\_if\\_metric\\_address\\_less\\_if](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int tos, int \*ifMetricAddrLessif, u\_int32\_t vrId)  
*Gets the addressless IF index for interfaces that do not have an IP address. This facilitates the creation of instances of addressed and address-less interfaces. This variable takes the value 0 on interfaces with IP Addresses, and the value of ifIndex for interfaces having no IP Address. On row creation, this value can be derived from the instance.*
- int [smi\\_ospf\\_get\\_if\\_metric\\_value](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int typeOfService, int \*ifMetricValue, u\_int32\_t vrId)  
*Gets the metric of using this TOS on this interface.*
- int [smi\\_ospf\\_get\\_if\\_metric\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int typeOfService, int \*ifMetricStatus, u\_int32\_t vrId)  
*Gets object permits management of the table by facilitating actions such as row creation, construction, and destruction.*
- int [smi\\_ospf\\_set\\_if\\_metric\\_value](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int typeOfService, int ifMetricValue, u\_int32\_t vrId)  
*Sets the metric of using this Type of Service on this interface.*
- int [smi\\_ospf\\_set\\_if\\_metric\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int typeOfService, int ifMetricStatus, u\_int32\_t vrId)  
*Sets object permits management of the table by facilitating actions such as row creation, construction, and destruction.*
- int [smi\\_ospf\\_get\\_virt\\_if\\_area\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, struct pal\_in4\_addr \*virtIfAreaId, u\_int32\_t vrId)  
*Gets the transit area that the virtual link traverses. By definition, this is not 0.0.0.0.*
- int [smi\\_ospf\\_get\\_virt\\_if\\_neighbor](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, struct pal\_in4\_addr \*virtIfNeighbor, u\_int32\_t vrId)

*Gets the router ID of the virtual neighbor.*

- int [smi\\_ospf\\_get\\_virt\\_if\\_transit\\_delay](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int \*virtIfTransitDelay, u\_int32\_t vrId)

*Gets the estimated number of seconds it takes to transmit a linkstate update packet over this interface.*

- int [smi\\_ospf\\_get\\_virt\\_if\\_retrans\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int \*virtIfRetransIntvl, u\_int32\_t vrId)

*Gets the retransmission interval time, in seconds, between link state advertisement retransmissions, for adjacencies belonging to this interface. This value is also used when retransmitting database description and Link State request packets. This value should be well over the expected round-trip time.*

- int [smi\\_ospf\\_get\\_virt\\_if\\_hello\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int \*virtIfHelloIntvl, u\_int32\_t vrId)

*Gets the hello interval time, in seconds, between the Hello packets the router sends on the interface. This value must be the same for the virtual neighbor.*

- int [smi\\_ospf\\_get\\_virt\\_if\\_rtr\\_dead\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int \*virtIfRouterDeadIntvl, u\_int32\_t vrId)

*Gets the dead interval time, in seconds, that a router's Hello packets have not been seen before its neighbors declare the router down. This should be some multiple of the Hello interval. This value must be the same for the virtual neighbor.*

- int [smi\\_ospf\\_get\\_virt\\_if\\_state](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int \*virtIfState, u\_int32\_t vrId)

*Gets the OSPF virtual interface states.*

- int [smi\\_ospf\\_get\\_virt\\_if\\_events](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int \*virtIfEvents, u\_int32\_t vrId)

*Gets the number of state changes or error events on this virtual link.*

- int [smi\\_ospf\\_get\\_virt\\_if\\_auth\\_type](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int \*virtIfAuthType, u\_int32\_t vrId)

*Gets the OSPF authentication type of this interface.*

- int [smi\\_ospf\\_get\\_virt\\_if\\_auth\\_key](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, char \*virtIfAuthKey, u\_int32\_t vrId)

*Gets the authentication key.*



- int [smi\\_ospf\\_get\\_virt\\_if\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int \*virtIfStatus, u\_int32\_t vrId)

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

- int [smi\\_ospf\\_set\\_virt\\_if\\_retrans\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int virtIfRetransIntvl, u\_int32\_t vrId)

*Sets the retransmission interval time, in seconds, between link state advertisement re-transmissions, for adjacencies belonging to this interface. This value is also used when retransmitting database description and Link State request packets. This value should be well over the expected round-trip time.*

- int [smi\\_ospf\\_set\\_virt\\_if\\_transit\\_delay](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtTransmitDelay, u\_int32\_t vrId)
- int [smi\\_ospf\\_set\\_virt\\_if\\_hello\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtHelloInterval, u\_int32\_t vrId)
- int [smi\\_ospf\\_set\\_virt\\_if\\_rtr\\_dead\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtDeadInterval, u\_int32\_t vrId)
- int [smi\\_ospf\\_set\\_virt\\_if\\_auth\\_type](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtAuthType, u\_int32\_t vrId)
- int [smi\\_ospf\\_set\\_virt\\_if\\_auth\\_key](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, char \*ifVirtAuthKey, u\_int32\_t vrId)
- int [smi\\_ospf\\_set\\_virt\\_if\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int virtIfStatus, u\_int32\_t vrId)

*Sets an object that permits management of the table facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_get\\_nbr\\_ip\\_addr](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr, int ifIndex, struct pal\_in4\_addr \*nbr\_ip\_addr, u\_int32\_t vrId)

*Gets the IP address this neighbor is using in its IP source address. Note that, on addressless links, this will not be 0.0.0.0 but the address of another of the neighbor's interfaces.*

- int [smi\\_ospf\\_get\\_nbr\\_address\\_less\\_index](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr, int ifIndex, int \*neighborIpAddrLessIndex, u\_int32\_t vrId)

*Gets the index, on an interface having an IP address, zero. On addressless interfaces, the corresponding value of ifIndex in the Internet Standard MIB. On row creation, this can be derived from the instance.*

- int [smi\\_ospf\\_get\\_nbr\\_rtr\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, struct pal\_in4\_addr \*nbrRouterId, u\_int32\_t vrId)

*Gets the 32-bit integer (represented as a type IPAddress) uniquely identifying the neighboring router in the Autonomous System.*

- int [smi\\_ospf\\_get\\_nbr\\_options](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int \*nbrOptions, u\_int32\_t vrId)

*Gets a bit mask corresponding to the neighbor's options field. Bit 0, if set, indicates the system operates on Type of Service metrics other than TOS 0. If zero, the neighbor ignores all metrics, except the TOS 0 metric. Bit 1, if set, indicates the associated area accepts and operates on external information; if zero, it is a stub area. Bit 2, if set, indicates the system routes IP Multicast datagrams, implementing the Multicast Extensions to OSPF. Bit 3, if set, indicates the associated area is an NSSA. These areas carry type 7 external advertisements, which they translate into type 5 external advertisements at NSSA borders.*

- int [smi\\_ospf\\_get\\_nbr\\_priority](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int \*neighborPriority, u\_int32\_t vrId)

*Gets the priority of this neighbor in the designated router election algorithm. The value 0 signifies that the neighbor is not eligible to become the designated router on this particular network.*

- int [smi\\_ospf\\_get\\_nbr\\_state](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int \*nbrState, u\_int32\_t vrId)

*Gets the state of the relationship with the neighbor.*

- int [smi\\_ospf\\_get\\_nbr\\_events](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int \*nbrEvents, u\_int32\_t vrId)

*Gets the number of times this neighbor relationship has changed state, or an error has occurred.*

- int [smi\\_ospf\\_get\\_nbr\\_ls\\_retrans\\_qlen](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int \*nbrLsRetransQlen, u\_int32\_t vrId)

*Gets the current length of the retransmission queue.*

- int [smi\\_ospf\\_get\\_nbma\\_nbr\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int \*nbmaNeighborStatus, u\_int32\_t vrId)

*Gets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_get\\_nbma\\_nbr\\_permanence](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int \*nbmaNeighborPermanence, u\_int32\_t vrId)

*Gets the variable that displays the status of the entry; 'dynamic' and 'permanent' refer to how the neighbor became known.*

- int [smi\\_ospf\\_get\\_nbr\\_hello\\_suppressed](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int \*nbrHelloSuppressed, u\_int32\_t vrId)

*Gets an indication whether Hellos are being suppressed to the neighbor.*

- int [smi\\_ospf\\_set\\_nbr\\_priority](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int neighborPriority, u\_int32\_t vrId)

*Sets the priority of this neighbor in the designated router election algorithm. The value 0 signifies that the neighbor is not eligible to become the designated router on this particular network.*

- int [smi\\_ospf\\_set\\_nbma\\_nbr\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int nbmaNeighborStatus, u\_int32\_t vrId)

*Sets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_get\\_virt\\_nbr\\_area](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr neighborIpAddress, struct pal\_in4\_addr \*virtNbrArea, u\_int32\_t vrId)

*Gets the transit area identifier.*

- int [smi\\_ospf\\_get\\_virt\\_nbr\\_rtr\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, struct pal\_in4\_addr \*virtNbrRouterId, u\_int32\_t vrId)

*Gets the 32-bit integer that uniquely identifies the neighboring router in the Autonomous System.*

- int [smi\\_ospf\\_get\\_virt\\_nbr\\_ip\\_addr](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, struct pal\_in4\_addr \*virtNbrIpAddress, u\_int32\_t vrId)

*Gets the IP address the virtual neighbor is using.*

- int [smi\\_ospf\\_get\\_virt\\_nbr\\_options](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, int \*virtNbrOptions, u\_int32\_t vrId)

*Gets the A bit mask corresponding to the neighbor's options field. Bit 1, if set, indicates that the system will operate on Type of Service metrics other than ToS 0. If zero, the neighbor will ignore all metrics except the TOS 0 metric. Bit 2, if set, indicates that the system is network multicast capable, i.e., that it implements OSPF multicast routing.*

- int [smi\\_ospf\\_get\\_virt\\_nbr\\_state](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, int \*virtNbrState, u\_int32\_t vrId)

*Gets the state of the virtual neighbor relationship.*

- int [smi\\_ospf\\_get\\_virt\\_nbr\\_events](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, int \*virtNbrEvents, u\_int32\_t vrId)

*Gets the number of times this virtual link has changed its state or an error has occurred.*

- int [smi\\_ospf\\_get\\_virt\\_nbr\\_ls\\_retrans\\_qlen](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, int \*virtNbrLsRetransQlen, u\_int32\_t vrId)

*Gets the current length of the retransmission queue.*

- int [smi\\_ospf\\_get\\_virt\\_nbr\\_hello\\_suppressed](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, int \*virtNbrHelloSuppressed, u\_int32\_t vrId)

*Gets an indication whether Hellos are being suppressed to the neighbor.*

- int [smi\\_ospf\\_get\\_ext\\_lsdb\\_type](#) (struct smiclient\_globals \*azg, int ospfProcessId, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*extLsdbType, u\_int32\_t vrId)

*Gets the LSA type. Each type has a distinct advertising format.*

- int [smi\\_ospf\\_get\\_ext\\_lsdb\\_lsid](#) (struct smiclient\_globals \*azg, int ospfProcessId, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, struct pal\_in4\_addr \*extLsdbLsid, u\_int32\_t vrId)

*Gets the link state ID. This ID is an LS Type Specific field containing either a Router ID or an IP Address; it identifies the piece of the routing domain that is being described by the advertisement.*

- int [smi\\_ospf\\_get\\_ext\\_lsdb\\_router\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, struct pal\_in4\_addr \*extLsdbRouterId, u\_int32\_t vrId)

*Gets the 32-bit number that uniquely identifies the originating router in the Autonomous System.*

- int [smi\\_ospf\\_get\\_ext\\_lsdb\\_sequence](#) (struct smiclient\_globals \*azg, int ospfProcessId, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*extLsdbSequence, u\_int32\_t vrId)

*Gets the sequence number field is a signed 32-bit integer. It starts with the value '80000001'h, or -'7FFFFFFF'h, and increments until '7FFFFFFF'h. Thus, a typical sequence number will be very negative. It is used to detect old and duplicate link state advertisements. The space of sequence numbers is linearly ordered. The larger the sequence number, the more recent the advertisement.*

- int [smi\\_ospf\\_get\\_ext\\_lsdb\\_age](#) (struct smiclient\_globals \*azg, int ospfProcessId, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*extLsdbAge, u\_int32\_t vrId)

*Gets the age of the LSA in seconds.*

- int [smi\\_ospf\\_get\\_ext\\_lsdbs\\_checksum](#) (struct smiclient\_globals \*azg, int ospf-ProcessId, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*extLsdbChecksum, u\_int32\_t vrId)

*Gets the 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. The checksum used is the same that is used for ISO connectionless datagrams; it is commonly referred to as the Fletcher checksum.*

- int [smi\\_ospf\\_get\\_ext\\_lsdbs\\_advertisement](#) (struct smiclient\_globals \*azg, int ospfProcessId, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, struct smi\_lsa\_header \*lsa, size\_t \*extLsdbAdv, u\_int32\_t vrId)

*Gets the entire link state advertisement, including its header.*

- int [smi\\_ospf\\_get\\_area\\_aggregate\\_area\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int type, struct pal\_in4\_addr addr, struct pal\_in4\_addr mask, struct pal\_in4\_addr \*areaAggrAreaId, u\_int32\_t vrId)

*Gets the area within which the address aggregate is to be found.*

- int [smi\\_ospf\\_get\\_area\\_aggregate\\_lsdbs\\_type](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int type, struct pal\_in4\_addr addr, struct pal\_in4\_addr mask, int \*areaAggrLsdbType, u\_int32\_t vrId)

*Gets type of the address aggregate. This field specifies the Lsdb type that this address aggregate applies to.*

- int [smi\\_ospf\\_get\\_area\\_aggregate\\_net](#) (struct smiclient\_globals \*azg, int ospf-ProcessId, struct pal\_in4\_addr areaId, int type, struct pal\_in4\_addr addr, struct pal\_in4\_addr mask, struct pal\_in4\_addr \*areaAggrNet, u\_int32\_t vrId)

*Gets the IP address of the net or subnet indicated by the range.*

- int [smi\\_ospf\\_get\\_area\\_aggregate\\_mask](#) (struct smiclient\_globals \*azg, int ospf-ProcessId, struct pal\_in4\_addr areaId, int type, struct pal\_in4\_addr addr, struct pal\_in4\_addr mask, struct pal\_in4\_addr \*areaAggrMask, u\_int32\_t vrId)

*Gets subnet mask that pertains to the net or subnet.*

- int [smi\\_ospf\\_get\\_area\\_aggregate\\_status](#) (struct smiclient\_globals \*azg, int ospf-ProcessId, struct pal\_in4\_addr areaId, int type, struct pal\_in4\_addr addr, struct pal\_in4\_addr mask, int \*areaAggrStatus, u\_int32\_t vrId)

*Gets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_get\\_area\\_aggregate\\_effect](#) (struct smiclient\_globals \*azg, int ospf-ProcessId, struct pal\_in4\_addr areaId, int type, struct pal\_in4\_addr addr, struct pal\_in4\_addr mask, int \*areaAggrEffect, u\_int32\_t vrId)

*Gets and object that tells, subnets subsumed by ranges either trigger the advertisement of the indicated aggregate (advertiseMatching) or result in the subnet's not being advertised at all outside the area.*

- [int smi\\_ospf\\_set\\_area\\_aggregate\\_status](#) (struct smiclient\_globals \*azg, int ospf-ProcessId, struct pal\_in4\_addr areaId, int addrAggrType, struct pal\_in4\_addr ipAddr, struct pal\_in4\_addr mask, int areaAggrStatus, u\_int32\_t vrId)

*Sets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction.*

- [int smi\\_ospf\\_set\\_area\\_aggregate\\_effect](#) (struct smiclient\_globals \*azg, int ospf-ProcessId, struct pal\_in4\_addr areaId, int addrAggrType, struct pal\_in4\_addr ipAddr, struct pal\_in4\_addr mask, int areaAggrEffect, u\_int32\_t vrId)

*Sets and object that tells, subnets subsumed by ranges either trigger the advertisement of the indicated aggregate (advertiseMatching) or result in the subnet's not being advertised at all outside the area.*

- [int smi\\_show\\_ospf\\_database\\_summary\\_info](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int16\_t ospf\_id, int start\_index, int end\_index, struct list \*dbSummaryList, int(\*funpointer)(struct list \*dbSummaryList))

*Fetches the OSPF process Database information summary.*

- [int smi\\_show\\_ospf\\_database\\_detail\\_info](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int16\_t ospf\_id, int start\_index, int end\_index, struct list \*dbDetailList, int(\*funpointer)(struct list \*dbDetailList))

*Fetches the OSPF process Database detailed information.*

- [int smi\\_show\\_ospf\\_interface\\_brief\\_info](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, int start\_index, int end\_index, struct list \*ifBriefList, int(\*funpointer)(struct list \*ifBriefList))

*Fetches the OSPF process interface related brief information.*

- [int smi\\_show\\_ospf\\_buffer\\_info](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int16\_t ospfProcessId, int start\_index, int end\_index, struct list \*bufList, int(\*funpointer)(struct list \*bufList))

*Fetches the OSPF Buffer statistics related information.*

- [int smi\\_show\\_ospf\\_borderrouter\\_info](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospf\_id, int start\_index, int end\_index, struct list \*brList, int(\*funpointer)(struct list \*brList))

*Fetches the OSPF Border routers related information.*

- [int smi\\_ospf\\_if\\_dna\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

*Unsets flood-reduction.*

- [int smi\\_show\\_ospf\\_nbr\\_info](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospf\_id, int start\_index, int end\_index, struct list \*nbrList, int(\*funpointer)(struct list \*nbrList))

*Fetches the OSPF Neighbors related information.*

- int **smi\_show\_ospf\_nbr\_detail** (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospf\_id, int start\_index, int end\_index, char \*vrf\_name, struct list \*ospfNbrDtlList, int(\*funpointer)(struct list \*ospfNbrDtlList))
- int **smi\_show\_ospf\_proc\_info** (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospf\_id, int start\_index, int end\_index, struct list \*processList, int(\*funpointer)(struct list \*processList))

*Fetches the OSPF process instance related information.*

- int **smi\_show\_ospf\_multiarea\_info** (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospf\_id, int start\_index, int end\_index, struct list \*multiAreaList, int(\*funpointer)(struct list \*multiAreaList))

*Fetches the OSPF multi area adjacencies configuration information.*

- int **smi\_show\_ospf\_vlink\_info** (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospf\_id, int start\_index, int end\_index, struct list \*vlinkList, int(\*funpointer)(struct list \*vlinkList))

*Fetches the OSPF Virtual links configuration information.*

- int **smi\_show\_if\_info** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, int start\_index, int end\_index, struct list \*ifInfoList, int(\*funpointer)(struct list \*ifInfoList))

*Fetches the OSPF process interface related information.*

- int **smi\_show\_ospf\_route\_summary\_info** (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int16\_t ospf\_id, int start\_index, int end\_index, struct list \*routeSumList, int(\*funpointer)(struct list \*routeSumList))

*Fetches the OSPF Route summary information.*

- int **smi\_show\_ospf\_route\_info** (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int16\_t ospf\_id, int start\_index, int end\_index, struct list \*routeList, int(\*funpointer)(struct list \*routeList))

*Fetches the OSPF Route information.*

- int **smi\_ospf\_show\_debugging** (struct smiclient\_globals \*azg, u\_int32\_t vrId, struct smi\_debug\_ospf \*smiDebug)
- int **smi\_ospf\_if\_passive\_interface\_unset** (struct smiclient\_globals \*azg, int vrId, char \*ifName)

*Sets This function unconfigure the passive interface.*

- int **smi\_ospf\_get\_virt\_local\_lsdb\_checksum** (struct smiclient\_globals \*azg, int proc, struct pal\_in4\_addr transit\_area, struct pal\_in4\_addr neighbor\_id, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*virt\_local\_lsdb\_checksum, u\_int32\_t vrId)

*Gets the 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. The checksum used is the same that is used for ISO connectionless datagrams; it is commonly referred to as the Fletcher checksum.*

- int [smi\\_ospf\\_graceful\\_restart\\_planned\\_set\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId)

*This function allows to configure only OSPF planned (S/W) restarts.*

- int [smi\\_ospf\\_get\\_local\\_lsdb\\_sequence](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr lsdbIp, signed int locLsdbAddLeesIf, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*localLsdbSequence, u\_int32\_t vrId)

*Gets the sequence number field is a signed 32-bit integer.*

*It starts with the value '80000001'h, or -'7FFFFFFF'h, and increments*

*until '7FFFFFFF'h. Thus, a typical sequence number will be very negative.*

*It is used to detect old and duplicate link state advertisements.*

*The space of sequence numbers is linearly ordered.*

*The larger the sequence number, the more recent the advertisement.*

- int [smi\\_ospf\\_get\\_restart\\_age](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*restartAge, u\_int32\_t vrId)

*Gets the remaining time in current OSPF graceful restart @ interval.*

- int [smi\\_ospf\\_get\\_virt\\_if\\_lsa\\_cksumsum](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, int \*virtLfLsaCksumsum, u\_int32\_t vrId)

*This function determine if there has been a change in virtual interfaces link state-database,.*

- int [smi\\_ospf\\_get\\_opaque\\_lsa\\_support](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*opaque\_lsaSupport, u\_int32\_t vrId)

*Gets the routers support for OSPF graceful restart.*

- int [smi\\_ospf\\_get\\_restart\\_support](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*restartSupport, u\_int32\_t vrId)

*Gets the routers support for OSPF graceful restart.*

- int [smi\\_ospf\\_area\\_default\\_cost\\_value\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_int32\_t areaCost)

*This function assigns the specified cost to the default summary route used for a not so stubby area (NSSA).*

- int [smi\\_ospf\\_get\\_nbr\\_restart\\_helper\\_exit\\_reason](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddress, int ifIndex, int \*nbrRestartHelperExitReason, u\_int32\_t vrId)

*signifies that there has been a change in the graceful restart helper state for the neighbour. This trap should be generated when the neighbour restart helper status transitions for a neighbour*

- int [smi\\_ospf\\_get\\_reference\\_bandwidth](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*referenceBandwidth, u\_int32\_t vrId)

*Gets the reference\_bandwidth in Kilobits/sec for default interface metrics.*



- int [smi\\_ospf\\_get\\_nbr\\_restart\\_helper\\_age](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr, int ifIndex, int \*restartHelperAge, u\_int32\_t vrId)  
*Gets the remaining time in current OSPF graceful restartinterval.*
- int [smi\\_ospf\\_get\\_virt\\_nbr\\_restart\\_helper\\_age](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, int \*virtNbrRestartHelperAge, u\_int32\_t vrId)  
*Gets the remaining time in current OSPF graceful restartinterval.*
- int [smi\\_ospf\\_graceful\\_restart\\_set\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int restartSeconds)  
*Sets the grace period in seconds for OSPF to restart gracefully along with reason for restart.*
- int [smi\\_ospf\\_hitless\\_restart\\_set\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int hitlessRestartSeconds)
- int [smi\\_ospf\\_get\\_restart\\_exit\\_reason](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*restartExitReason, u\_int32\_t vrId)  
*Gets the outcome of the last attempt at a restart.*
- int [smi\\_ospf\\_if\\_ip\\_router\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr areaId, int areaFormat, int ospfProcessId, int ospfInterfaceInstanceId, char \*cmdOptionalString)  
*Sets This function enable the OSPF on interface.*
- int [smi\\_ospf\\_log\\_adj\\_changes\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*logAdjacencyDebugType)  
*Sets This function configure the log adjacency.*
- int [smi\\_ospf\\_get\\_settrap](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*trapFlagBit, u\_int32\_t vrId)  
*Gets the bitmap of traps in ospf. every bit indicates a trap in ospf.*
- int [smi\\_ospf\\_if\\_ip\\_router\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr areaId, int areaFormat, int ospfProcessId, int ospfInterfaceInstanceId)  
*Sets This function removes OSPF on interface.*
- int [smi\\_ospf\\_get\\_discontinuity\\_time](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*retDiscontinuityTime, u\_int32\_t vrId)  
*Gets the value of sysUpTime on the most recent occasion which any one of this MIB.s counters suffered a discontinuity.*
- int [smi\\_ospf\\_get\\_virt\\_if\\_lsa\\_count](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr neighborIpAddr, int \*virtiflsaCount, u\_int32\_t vrId)

*This function gets the total number of link-local link state this virtual interfaces link-local link state database.*

- int [smi\\_ospf\\_get\\_as\\_lsdb\\_checksum](#) (struct smiclient\_globals \*azg, int ospfProcessId, int lsaType, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*as\_lsdb\_checksum, u\_int32\_t vrId)

*Gets the 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. The checksum used is the same that is used for ISO connectionless datagrams; it is commonly referred to as the Fletcher checksum.*

- int [smi\\_ospf\\_get\\_inter\\_area\\_metric](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*inteAreaMetric, u\_int32\_t vrId)

*Gets the inter area metric value.*

- int [smi\\_ospf\\_get\\_area\\_lsa\\_count\\_number](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \*areaLsaCountNumber, u\_int32\_t vrId)

*Gets the total number of link state advertisements in this area's link state database, excluding AS-external LSAs.*

- int [smi\\_ospf\\_get\\_as\\_lsdb\\_age](#) (struct smiclient\_globals \*azg, int ospfProcessId, int lsaType, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*as\_lsdb\_age, u\_int32\_t vrId)

*Gets the age of the LSA in seconds.*

- int [smi\\_ospf\\_max\\_area\\_limit\\_set\\_sdkapi](#) (struct smiclient\_globals \*azg, int ospfProcessId, u\_int32\_t vrId, u\_int32\_t maxAreaLimit)

*Sets the maximum number of OSPF areas.*

- int [smi\\_ospf\\_get\\_as\\_lsdb\\_sequence](#) (struct smiclient\_globals \*azg, int ospfProcessId, int lsaType, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*as\_lsdb\_sequence, u\_int32\_t vrId)

*Gets the sequence number field is a signed 32-bit integer. It starts with the value '80000001'h, or -'7FFFFFFF'h, and increments until '7FFFFFFF'h. Thus, a typical sequence number will be very negative. It is used to detect old and duplicate link state advertisements. The space of sequence numbers is linearly ordered. The larger the sequence number, the more recent the advertisement.*

- int [smi\\_ospf\\_get\\_restart\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*restartStatus, u\_int32\_t vrId)

*Gets the Current status of OSPF graceful restart.*

- int [smi\\_ospf\\_log\\_adj\\_changes\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*logAdjacencyDebugType)

*Sets This function unconfigure the log adjacency.*

- int [smi\\_ospf\\_get\\_nbr\\_restart\\_helper\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr, int ifIndex, int \*nbrRestartHelperStatus, u\_int32\_t vrId)  
*Gets indication whether the router is acting as a graceful restart helper for the neighbor.*
- int [smi\\_ospf\\_if\\_passive\\_interface\\_set](#) (struct smiclient\_globals \*azg, int vrId, char \*ifName)  
*Sets This function configures the passive interface.*
- int [smi\\_ospf\\_set\\_settrap](#) (struct smiclient\_globals \*azg, int ospfProcessId, int trapFlagBit, u\_int32\_t vrId)  
*Sets the trap bit for a particular trap in ospf.*
- int [smi\\_ospf\\_get\\_intra\\_area\\_metric](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*intraAreaMetric, u\_int32\_t vrId)  
*Gets the inter area metric value.*
- int [smi\\_ospf\\_get\\_local\\_lsdb\\_checksum](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr lsdbIp, signed int locLsdbAddLeesIf, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*localLsdbChecksum, u\_int32\_t vrId)  
*Gets the 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. The checksum used is the same that is used for ISO connectionless datagrams; it is commonly referred to as the Fletcher checksum.*
- int [smi\\_ospf\\_get\\_stub\\_router\\_support](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*as\_scopeLsaChecksum, u\_int32\_t vrId)  
*Gets the flag to note whether this router supports demand routing.*
- int [smi\\_ospf\\_restart\\_graceful\\_sdkapi](#) (struct smiclient\_globals \*azg, int restartSeconds, int restartReason, u\_int32\_t vrId)  
*Sets the grace period in seconds for OSPF to restart gracefully along with reason for restart.*
- int [smi\\_ospf\\_get\\_as\\_scope\\_lsa\\_count](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*asScopeLsaCount, u\_int32\_t vrId)  
*Gets the number of AS-scope link state in the AS-scope link state database.*
- int [smi\\_ospf\\_get\\_stub\\_router\\_advertisement](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*stubRouterAdvertisement, u\_int32\_t vrId)  
*Gets the number of AS-scope link state in the AS-scope link state database.*
- int [smi\\_ospf\\_timers\\_spf\\_validate\\_and\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t startDelay, u\_int32\_t minDelay, u\_int32\_t maxDelay)

*Sets This function sets the minimum and maximum delay between a topology change, being either received in an LSA or self detected, and the SPF calculation being run.*

- int [smi\\_ospf\\_graceful\\_restart\\_unset\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId)

*Disables the OSPF graceful restart method and sets grace period to zero.*

- int [smi\\_ospf\\_hitless\\_restart\\_unset\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId)
- int [smi\\_ospf\\_get\\_external\\_type2\\_metric](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*externalType2Metric, u\_int32\_t vrId)

*Gets the inter area metric value.*

- int [smi\\_ospf\\_get\\_restart\\_strict\\_lsa\\_check](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*restartStrictLsaCheck, u\_int32\_t vrId)

*Gets the strict LSA checking is enabled for restart.*

- int [smi\\_ospf\\_get\\_restart\\_interval](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*time, u\_int32\_t vrId)

*Gets the OSPF graceful restart timeout interval.*

- int [smi\\_ospf\\_process\\_shut\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospfProcessId)

*This function unconfigures the ip ospf shutdown command.*

- int [smi\\_ospf\\_get\\_virt\\_local\\_lsdb\\_age](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr transitArea, struct pal\_in4\_addr neighborIpAddr, int lsaType, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*virtLocalLsdbAge, u\_int32\_t vrId)

*Gets the age of the LSA in seconds.*

- int [smi\\_ospf\\_process\\_shut\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospfProcessId)

*This function configures the ip ospf shutdown command.*

- int [smi\\_ospf\\_vlink\\_retransmit\\_interval\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int retransmitInterval)

*This function sets the retransmit interval value for the virtual interface.*

- int [smi\\_ospf\\_get\\_local\\_lsdb\\_age](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr lsdbIp, signed int locLsdbLddLeesIf, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*localLsdbAge, u\_int32\_t vrId)

*Gets the age of the LSA in seconds.*

- int [smi\\_ospf\\_graceful\\_restart\\_planned\\_unset\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId)

*This function allows to configure all types of restarts (Planned and Unplanned).*

- int [smi\\_ospf\\_get\\_compatible\\_rfc1583](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*compatibleRfc1583, u\_int32\_t vrId)

*Gets the whether RFC1583Compatibility is enabled or not. default interface metrics.*

- int [smi\\_ospf\\_get\\_virt\\_local\\_lsdb\\_sequence](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr transitArea, struct pal\_in4\_addr neighborId, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \*virtLocalSsdbSequence, u\_int32\_t vrId)

*Gets the sequence number field is a signed 32-bit integer. It starts with the value '80000001'h, or -'7FFFFFFF'h, and increments until '7FFFFFFF'h. Thus, a typical sequence number will be very negative. It is used to detect old and duplicate link state advertisements. The space of sequence numbers is linearly ordered. The larger the sequence number, the more recent the advertisement.*

- int [smi\\_ospf\\_auto\\_cost\\_reference\\_bandwidth\\_type\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int referenceBandwidth, char \*bandwidthType)

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

- int [smi\\_ospf\\_area\\_auth\\_by\\_type\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char authType)

*This function disables the authentication type for the area.*

- int [smi\\_ospf\\_area\\_nssa\\_default\\_originate\\_route\\_map\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, char \*routeMapName)

*Sets the route-map for the routes originated from this NSSA router.*

- int [smi\\_ospf\\_set\\_lsdb\\_limit\\_sdkapi](#) (struct smiclient\_globals \*azg, int ospfProcessId, u\_int32\_t vrId, u\_int32\_t lsdbLimit, int actionType, int isLsdbLimit)

*Sets the the maximum number of LSAs that can be supported by the OSPF instance, along with action (Hard/Soft) to performed in case the number of LSAs exceeds the specified limit.*

- int [smi\\_ospf\\_set\\_area\\_aggregate\\_route\\_tag](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int addrAggrType, struct pal\_in4\_addr ipAddr, struct pal\_in4\_addr mask, u\_int32\_t routeTag, u\_int32\_t vrId)

*Sets and object that tells, subnets subsumed by ranges either trigger the advertisement of the indicated aggregate (advertiseMatching) or result in the subnet's not being advertised at all outside the area.*

- int [smi\\_ospf\\_get\\_external\\_type1\\_metric](#) (struct smiclient\_globals \*azg, int ospfProcessId, int \*externalType1Metric, u\_int32\_t vrId)

*Gets the inter area metric value.*

- int [smi\\_ospf\\_get\\_host\\_cfg\\_area\\_id](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr addr, int tos, struct pal\_in4\_addr \*hostCfgAreaId, u\_int32\_t vrId)

*Gets the OSPF area to which the host belongs. facilitating actions such as row creation, construction, and destruction.*

- int [smi\\_ospf\\_get\\_if\\_lsa\\_count](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr addr, int ifIndex, int \*if\_lsa\_count, u\_int32\_t vrId)

*Gets the LSA count for an interface.*

- int [smi\\_ospf\\_get\\_if\\_lsa\\_checksum](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr addr, int ifIndex, int \*ifLsaChecksum, u\_int32\_t vrId)

*Gets the get the checksum of the complete contents of the advertisement, excepting the age field.*

- int [smi\\_ospf\\_get\\_if\\_dr](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr addr, int ifIndex, struct pal\_in4\_addr \*ifDr, u\_int32\_t vrId)

*Gets the get the DR id.*

- int [smi\\_ospf\\_get\\_if\\_bdr](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr addr, int ifIndex, struct pal\_in4\_addr \*ifBdr, u\_int32\_t vrId)

*Gets the get the DR id.*

- int [smi\\_ospf\\_get\\_virt\\_nbr\\_restart\\_helper\\_status](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, int \*virtNbrRestartHelperStatus, u\_int32\_t vrId)

- int [smi\\_ospf\\_get\\_virt\\_nbr\\_restart\\_helper\\_exit\\_reason](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr nbrId, int \*virtNbrRestartHelperExitReason, u\_int32\_t vrId)

- int [smi\\_ospf\\_get\\_area\\_aggregate\\_route\\_tag](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int type, struct pal\_in4\_addr addr, struct pal\_in4\_addr mask, int \*areaAggregateRouteTag, u\_int32\_t vrId)

*Gets the external route tag to be included in NSSA (type-7).*

- int [smi\\_ospf\\_if\\_cost\\_value\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifCost)

*Sets This function resets the cost for the current interface to the default value 10.*

- int [smi\\_ospf\\_summary\\_address\\_not\\_advertise\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask)

*This function sets the flag of the external summary address range to Not Advertise.*

- int [smi\\_ospf\\_max\\_area\\_limit\\_unset\\_sdkapi](#) (struct smiclient\_globals \*azg, int ospfProcessId, u\_int32\_t vrId)

*Removes the maximum number of OSPF areas that was set.*

- int [smi\\_ospf\\_set\\_if\\_type](#) (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifType, u\_int32\_t vrId)  
*Sets the OSPF interface type. By way of a default, this field may be intuited from the corresponding value of ifType. Broadcast LANs, such as Ethernet and IEEE 802.5, take the value 'broadcast', X.25 and similar technologies take the value 'nbma', and links that are definitively point to point take the value 'pointToPoint'.*
- int [smi\\_ospf\\_debug\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int debug)  
*Use this function to specify debugging option for OSPF ZebOS information.*
- int [smi\\_ospf\\_debug\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int debug)  
*Use this function to disable debugging option for OSPF ZebOS information.*
- int [smi\\_debug\\_ospf\\_packet\\_set](#) (struct smiclient\_globals \*azg, int vrId, int packetType, int debugMode, int debugDetail)  
*Use this function to specify the packet debugging options for OSPF ZebOS information.*
- int [smi\\_debug\\_ospf\\_packet\\_unset](#) (struct smiclient\_globals \*azg, int vrId, int packetType, int debugMode, int debugDetail)  
*Use this function to disable the packet debugging options for OSPF ZebOS information.*
- int [smi\\_ospf\\_process\\_clear](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int [smi\\_ospf\\_host\\_entry\\_cost\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr hostIpAddr, struct pal\_in4\_addr areaId)  
*This function unsets stub host entry belonging to particular area.*
- int [smi\\_ospf\\_te\\_link\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName)  
*Set detail of TE\_LINK.*
- int [smi\\_ospf\\_te\\_link\\_detail\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName, char \*ifName, int TELinkType)  
*Set detail of TE\_LINK.*
- int [smi\\_ospf\\_disable\\_ext\\_multi\\_inst](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId)  
*This call implements the no parameter of the enable ext-ospf-multi-inst command to disable support of multiple OSPF instances.*
- int [smi\\_ospf\\_if\\_dna\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)  
*Sets flood-reduction.*

- int [smi\\_ospf\\_dna\\_unset\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*Unset DNA.*
- int [smi\\_ospf\\_dna\\_set\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)  
*Set DNA.*
- int [smi\\_ospf\\_if\\_conf\\_ldp\\_igp\\_unset\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)  
*Unsets LDP-OSPF.*
- int [smi\\_ospf\\_if\\_conf\\_ldp\\_igp\\_set\\_sdkapi](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t holdDownTimer)  
*Sets LDP-OSPF.*
- int [smi\\_ospf\\_if\\_dna\\_unset\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int [smi\\_ospf\\_dna\\_unset\\_sdkapi\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int [smi\\_ospf\\_dna\\_set\\_sdkapi\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int [smi\\_ospf\\_if\\_conf\\_ldp\\_igp\\_unset\\_sdkapi\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int [smi\\_ospf\\_if\\_conf\\_ldp\\_igp\\_set\\_sdkapi\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t holddownTimer)
- int [smi\\_ospf\\_host\\_entry\\_cost\\_unset\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr hostIpAddr, struct pal\_in4\_addr areaId)
- int [smi\\_ospf\\_redist\\_metric\\_type\\_unset\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)
- int [smi\\_ospf\\_if\\_conf\\_ldp\\_igp\\_sync\\_sdkapi\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t holddownTimer)
- int [smi\\_ospf\\_interface\\_tunnel\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t tid)
- int [smi\\_ospf\\_process\\_set\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int [smi\\_ospf\\_process\\_set\\_vrf\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*vrfName)
- int [smi\\_ospf\\_te\\_link\\_enable\\_sdkapi\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*teLinkName)
- int [smi\\_ospf\\_process\\_unset\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int [smi\\_ospf\\_network\\_set\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr ospfNetAddr, u\_char netMask, struct pal\_in4\_addr areaId, s\_int16\_t ospfInterfaceInstanceId)



- int **smi\_ospf\_network\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr ospfNetAddr, u\_char netMask, struct pal\_in4\_addr areaId, s\_int16\_t ospfInterfaceInstanceId)
- int **smi\_ospf\_domain\_id\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*domainType, int domainValue, bool\_t primaryDomainId)
- int **smi\_ospf\_domain\_id\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*domainType, u\_int8\_t \*domainValue, bool\_t isPrimaryDomainId)
- int **smi\_ospf\_null\_domain\_id\_set\_sdkapi\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, int proc\_id, bool\_t nullDomainValue)
- int **smi\_ospf\_router\_id\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr routerId)
- int **smi\_ospf\_router\_id\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_passive\_interface\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*ifName)
- int **smi\_ospf\_passive\_interface\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_passive\_interface\_default\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_passive\_interface\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*ifName)
- int **smi\_ospf\_passive\_interface\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_passive\_interface\_default\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_host\_entry\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr ipAddr, struct pal\_in4\_addr areaId)
- int **smi\_ospf\_host\_entry\_unset\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr ipAddr, char \*areaId)
- int **smi\_ospf\_host\_entry\_cost\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr hostIpAddr, struct pal\_in4\_addr areaId, int hostEntryCost)
- int **smi\_ospf\_abr\_type\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_char areaBorderType)
- int **smi\_ospf\_abr\_type\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_compatible\_rfc1583\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_compatible\_rfc1583\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_timers\_spf\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t spfMinDelay, u\_int32\_t spfMaxDelay)

- int **smi\_ospf\_timers\_spf\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_timers\_refresh\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int refreshInterval)
- int **smi\_ospf\_timers\_refresh\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_lsa\_throttle\_timers\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t lsaTTStartDelay, u\_int32\_t holdInterval, u\_int32\_t lsaTTMaxDelay)
- int **smi\_ospf\_lsa\_throttle\_timers\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_lsa\_min\_arrival\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t lsaMinArrival)
- int **smi\_ospf\_lsa\_min\_arrival\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_auto\_cost\_reference\_bandwidth\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int referenceBandwidth)
- int **smi\_ospf\_auto\_cost\_reference\_bandwidth\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_max\_concurrent\_dd\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int16\_t maxDBDescriptor)
- int **smi\_ospf\_max\_concurrent\_dd\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_max\_unuse\_packet\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t maxPackets)
- int **smi\_ospf\_max\_unuse\_packet\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_max\_unuse\_lsa\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t maxUnuseLsa)
- int **smi\_ospf\_max\_unuse\_lsa\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_overflow\_database\_external\_limit\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t lsdBLimit)
- int **smi\_ospf\_overflow\_database\_external\_limit\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_overflow\_database\_external\_interval\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int exitOverflowInterval)
- int **smi\_ospf\_overflow\_database\_external\_interval\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_enable\_ext\_multi\_inst\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId)
- int **smi\_ospf\_if\_network\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, int ifNetworkType)
- int **smi\_ospf\_if\_network\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

- int **smi\_ospf\_if\_network\_p2mp\_nbma\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_authentication\_type\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_char ifAuthType)
- int **smi\_ospf\_if\_authentication\_type\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_char ifAuthType)
- int **smi\_ospf\_if\_authentication\_type\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_authentication\_type\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_priority\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_char ifPriority)
- int **smi\_ospf\_if\_priority\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_char ifPriority)
- int **smi\_ospf\_if\_priority\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_priority\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_mtu\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int16\_t ifMtu)
- int **smi\_ospf\_if\_mtu\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_mtu\_ignore\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_mtu\_ignore\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_mtu\_ignore\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_mtu\_ignore\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_cost\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifCost)
- int **smi\_ospf\_if\_cost\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifCost)
- int **smi\_ospf\_if\_cost\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_cost\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_transmit\_delay\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifTransmitDelay)
- int **smi\_ospf\_if\_transmit\_delay\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifTransmitDelay)
- int **smi\_ospf\_if\_transmit\_delay\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

- int **smi\_ospf\_if\_transmit\_delay\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_retransmit\_interval\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifRetransmitInterval)
- int **smi\_ospf\_if\_retransmit\_interval\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifRetransmitInterval)
- int **smi\_ospf\_if\_retransmit\_interval\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_retransmit\_interval\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_hello\_interval\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifHelloInterval)
- int **smi\_ospf\_if\_hello\_interval\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifHelloInterval)
- int **smi\_ospf\_if\_hello\_interval\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_hello\_interval\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_dead\_interval\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifDeadInterval)
- int **smi\_ospf\_if\_dead\_interval\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifDeadInterval)
- int **smi\_ospf\_if\_dead\_interval\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_dead\_interval\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_authentication\_key\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, char \*md5, char \*ifAuthKey)
- int **smi\_ospf\_if\_authentication\_key\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, char \*md5, char \*ifAuthKey)
- int **smi\_ospf\_if\_authentication\_key\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_authentication\_key\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_message\_digest\_key\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_char msgDigestKeyId, char \*md5, char \*ifAuthKey)
- int **smi\_ospf\_if\_message\_digest\_key\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_char msgDigestKeyId, char \*md5, char \*ifAuthKey)
- int **smi\_ospf\_if\_message\_digest\_key\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_char msgDigestKeyId)

- int **smi\_ospf\_if\_message\_digest\_key\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_char msgDigestKeyId)
- int **smi\_ospf\_if\_te\_metric\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifTEMetric)
- int **smi\_ospf\_if\_te\_metric\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_database\_filter\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_database\_filter\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_database\_filter\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_database\_filter\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_if\_disable\_all\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_disable\_all\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_resync\_timeout\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, u\_int32\_t ifResyncTimeout)
- int **smi\_ospf\_if\_resync\_timeout\_set\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifResyncTimeout)
- int **smi\_ospf\_if\_resync\_timeout\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int **smi\_ospf\_if\_resync\_timeout\_unset\_by\_addr\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr ipAddr)
- int **smi\_ospf\_vlink\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)
- int **smi\_ospf\_vlink\_unset\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, struct pal\_in4\_addr peerId)
- int **smi\_ospf\_vlink\_dead\_interval\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtDeadInterval)
- int **smi\_ospf\_vlink\_hello\_interval\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtHelloInterval)
- int **smi\_ospf\_vlink\_transmit\_delay\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtTransmitDelay)
- int **smi\_ospf\_vlink\_dead\_interval\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)
- int **smi\_ospf\_vlink\_hello\_interval\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)

- int **smi\_ospf\_vlink\_retransmit\_interval\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)
- int **smi\_ospf\_vlink\_transmit\_delay\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)
- int **smi\_ospf\_vlink\_authentication\_type\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtAuthType)
- int **smi\_ospf\_vlink\_authentication\_type\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)
- int **smi\_ospf\_vlink\_authentication\_key\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, char \*ifVirtAuthKey)
- int **smi\_ospf\_vlink\_authentication\_key\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)
- int **smi\_ospf\_vlink\_message\_digest\_key\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, u\_char virtMsgDigestKeyId, char \*ifVirtAuthKey)
- int **smi\_ospf\_vlink\_message\_digest\_key\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, u\_char virtMsgDigestKeyId)
- int **smi\_ospf\_summary\_address\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask)
- int **smi\_ospf\_summary\_address\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask)
- int **smi\_ospf\_summary\_address\_tag\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*summaryAddr, u\_int32\_t addressTag)
- int **smi\_ospf\_summary\_address\_tag\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask)
- int **smi\_ospf\_summary\_address\_not\_advertise\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask)
- int **smi\_ospf\_nbr\_static\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr)
- int **smi\_ospf\_nbr\_static\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr)
- int **smi\_ospf\_nbr\_static\_priority\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr, u\_char neighborStaticPriority)
- int **smi\_ospf\_nbr\_static\_priority\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr)

- **int smi\_ospf\_nbr\_static\_poll\_interval\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr, int nbrPollInterval)
- **int smi\_ospf\_nbr\_static\_poll\_interval\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr)
- **int smi\_ospf\_nbr\_static\_cost\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr, u\_int16\_t neighborCost)
- **int smi\_ospf\_nbr\_static\_cost\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr)
- **int smi\_ospf\_area\_auth\_type\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char areaAuthType)
- **int smi\_ospf\_area\_auth\_type\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- **int smi\_ospf\_area\_stub\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- **int smi\_ospf\_area\_stub\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- **int smi\_ospf\_area\_no\_summary\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- **int smi\_ospf\_area\_no\_summary\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- **int smi\_ospf\_area\_default\_cost\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_int32\_t areaCost)
- **int smi\_ospf\_area\_default\_cost\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- **int smi\_ospf\_area\_range\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, u\_char rangeMask)
- **int smi\_ospf\_area\_range\_unset\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, char \*ospfAreaRange)
- **int smi\_ospf\_area\_range\_not\_advertise\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, u\_char rangeMask)
- **int smi\_ospf\_area\_range\_not\_advertise\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, u\_char rangeMask)
- **int smi\_ospf\_area\_range\_substitute\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, char \*ospfAreaRange, char \*substituteAddr)
- **int smi\_ospf\_area\_range\_substitute\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, char \*ospfAreaRange)
- **int smi\_ospf\_area\_filter\_list\_prefix\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int prefixType, char \*prefixName)

- int **smi\_ospf\_area\_filter\_list\_prefix\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int prefixType)
- int **smi\_ospf\_area\_filter\_list\_access\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int accessType, char \*accessName)
- int **smi\_ospf\_area\_filter\_list\_access\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int accessType)
- int **smi\_ospf\_area\_export\_list\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, char \*accessListName)
- int **smi\_ospf\_area\_export\_list\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- int **smi\_ospf\_area\_import\_list\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, char \*accessListName)
- int **smi\_ospf\_area\_import\_list\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- int **smi\_ospf\_area\_shortcut\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char areaShortcutType)
- int **smi\_ospf\_area\_shortcut\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- s\_int32\_t **smi\_ospf\_multi\_area\_adjacency\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, char \*ifName, struct pal\_in4\_addr neighborIpAddr, int areaFormat)
- s\_int32\_t **smi\_ospf\_multi\_area\_adjacency\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char \*ifName, struct pal\_in4\_addr neighborIpAddr)
- int **smi\_ospf\_set\_nssa\_stability\_interval\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int nssaStabilityInterval, u\_int32\_t vrId)
- int **smi\_ospf\_area\_nssa\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- int **smi\_ospf\_area\_nssa\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- int **smi\_ospf\_area\_nssa\_translator\_role\_set\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, u\_char nssaTranslatorRole, int isNssaArea)
- int **smi\_ospf\_area\_nssa\_translator\_role\_set\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, u\_char nssaTranslatorRole, int isNssaArea)
- int **smi\_ospf\_area\_nssa\_translator\_role\_unset\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int isNssaArea)
- int **smi\_ospf\_area\_nssa\_translator\_role\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- int **smi\_ospf\_area\_nssa\_no\_redistribution\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)



- int **smi\_ospf\_area\_nssa\_no\_redistribution\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- int **smi\_ospf\_area\_nssa\_default\_originate\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- int **smi\_ospf\_area\_nssa\_default\_originate\_metric\_set\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int nssaDefaultOriginMetric, int isNssaArea)
- int **smi\_ospf\_area\_nssa\_default\_originate\_metric\_type\_set\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int nssaDefaultOriginMetricType, int isNssaArea)
- int **smi\_ospf\_area\_nssa\_default\_originate\_metric\_set\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int nssaDefaultOriginMetric, int isNssaArea)
- int **smi\_ospf\_area\_nssa\_default\_originate\_metric\_type\_set\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, int nssaDefaultOriginMetricType, int isNssaArea)
- int **smi\_ospf\_area\_nssa\_default\_originate\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)
- int **smi\_ospf\_area\_nssa\_stability\_interval\_set\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, u\_int32\_t nssaStabilityInterval, int isNssaArea)
- int **smi\_ospf\_area\_nssa\_stability\_interval\_set\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*areaId, u\_int32\_t nssaStabilityInterval, int isNssaArea)
- int **smi\_ospf\_opaque\_link\_lsa\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr addr, u\_char opaqueType, u\_int32\_t opaqueId, char \*opaqueData, u\_int32\_t opaqueLen)
- int **smi\_ospf\_opaque\_area\_lsa\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char opaqueType, u\_int32\_t opaqueId, char \*opaqueData, u\_int32\_t opaqueLen)
- int **smi\_ospf\_opaque\_as\_lsa\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_char opaqueType, u\_int32\_t opaqueId, char \*opaqueData, u\_int32\_t opaqueLen)
- int **smi\_ospf\_capability\_opaque\_lsa\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_capability\_opaque\_lsa\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_capability\_traffic\_engineering\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_capability\_traffic\_engineering\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_te\_link\_flood\_scope\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName, int ospfProcessId, struct pal\_in4\_addr areaId, int areaFormat)
- int **smi\_ospf\_te\_link\_flood\_scope\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName, int ospfProcessId, struct pal\_in4\_addr areaId)
- int **smi\_ospf\_telink\_te\_metric\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName, u\_int32\_t teMetric)

- **int smi\_ospf\_telink\_te\_metric\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName)
- **int smi\_ospf\_opaque\_te\_link\_local\_lsa\_enable\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName)
- **int smi\_ospf\_opaque\_te\_link\_local\_lsa\_disable\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*TELinkName)
- **int smi\_ospf\_capability\_cspf\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_capability\_cspf\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_cspf\_better\_protection\_type\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, bool\_t cspfProtectionType)
- **int smi\_ospf\_enable\_db\_summary\_opt\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_disable\_db\_summary\_opt\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_redistribute\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int secondaryOspfProcessId, int routeSourceType, int redistMetricType, int redistMetricValue)
- **int smi\_ospf\_redistribute\_default\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int sourceOrigin, int redistMetricType, int redistMetricValue)
- **int smi\_ospf\_redist\_proto\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)
- **int smi\_ospf\_redist\_proto\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)
- **int smi\_ospf\_redist\_default\_always\_set\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, bool\_t defaultAlways)
- **int smi\_ospf\_redist\_default\_always\_set\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, bool\_t defaultAlways)
- **int smi\_ospf\_redist\_default\_metric\_type\_set\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int defaultMetricType)
- **int smi\_ospf\_redist\_default\_metric\_type\_set\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int defaultMetricType)
- **int smi\_ospf\_redist\_default\_metric\_type\_unset\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_redist\_default\_metric\_type\_unset\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_redist\_metric\_type\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int redistMetricType, int secondaryOspfProcessId)
- **int smi\_ospf\_redist\_default\_metric\_set\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int defaultMetric)
- **int smi\_ospf\_redist\_default\_metric\_set\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int defaultMetric)

- int **smi\_ospf\_redist\_default\_metric\_unset\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_redist\_default\_metric\_unset\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_redist\_metric\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int redistMetricValue, int secondaryOspfProcessId)
- int **smi\_ospf\_redist\_metric\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)
- int **smi\_ospf\_redist\_tag\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, u\_int32\_t redistTag, int secondaryOspfProcessId)
- int **smi\_ospf\_redist\_tag\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)
- int **smi\_ospf\_redist\_default\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceOrigin)
- int **smi\_ospf\_redist\_default\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_distribute\_list\_out\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId, char \*accessListName)
- int **smi\_ospf\_distribute\_list\_out\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId, char \*accessListName)
- int **smi\_ospf\_distribute\_list\_in\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*accessListName)
- int **smi\_ospf\_distribute\_list\_in\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*accessListName)
- int **smi\_ospf\_routemap\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, char \*routeMapName, int secondaryOspfProcessId)
- int **smi\_ospf\_routemap\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)
- int **smi\_ospf\_routemap\_default\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*routeMapName)
- int **smi\_ospf\_routemap\_default\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_default\_metric\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int metricValue)
- int **smi\_ospf\_default\_metric\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- int **smi\_ospf\_distance\_all\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int adminDistance)
- int **smi\_ospf\_distance\_all\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)

- **int smi\_ospf\_distance\_intra\_area\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int intraAreaDistance)
- **int smi\_ospf\_distance\_intra\_area\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_distance\_inter\_area\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int interAreaDistance)
- **int smi\_ospf\_distance\_inter\_area\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_distance\_external\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int externalDistance)
- **int smi\_ospf\_distance\_external\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_distance\_source\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t adminDistance, struct pal\_in4\_addr sourceIpAddr, u\_char masklen, char \*accessListName)
- **int smi\_ospf\_distance\_source\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr sourceIpAddr, u\_char masklen, char \*accessListName)
- **int smi\_ospf\_capability\_restart\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int restartMethod)
- **int smi\_ospf\_capability\_restart\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_restart\_helper\_policy\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int restartHelperPolicy)
- **int smi\_ospf\_restart\_helper\_policy\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId)
- **int smi\_ospf\_restart\_helper\_never\_router\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, struct pal\_in4\_addr nbrRouterId)
- **int smi\_ospf\_restart\_helper\_never\_router\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, struct pal\_in4\_addr nbrRouterId)
- **int smi\_ospf\_restart\_helper\_never\_router\_unset\_all\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId)
- **int smi\_ospf\_restart\_helper\_grace\_period\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int restartHelperPeriod)
- **int smi\_ospf\_restart\_helper\_grace\_period\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId)
- **int smi\_ospf\_set\_asbdr\_rtr\_status\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, int areaASBDRRouterStatus, u\_int32\_t vrId)
- **int smi\_ospf\_set\_area\_status\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, int areaStatus, u\_int32\_t vrId)
- **int smi\_ospf\_set\_if\_area\_id\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, struct pal\_in4\_addr areaId, u\_int32\_t vrId)
- **int smi\_ospf\_set\_if\_admin\_stat\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int status, u\_int32\_t vrId)
- **int smi\_ospf\_set\_if\_rtr\_priority\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifRouterPriority, u\_int32\_t vrId)

- int **smi\_ospf\_set\_if\_transit\_delay\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifTransmitDelay, u\_int32\_t vrId)
- int **smi\_ospf\_set\_if\_retrans\_interval\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifRetransmitInterval, u\_int32\_t vrId)
- int **smi\_ospf\_set\_if\_hello\_interval\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifHelloInterval, u\_int32\_t vrId)
- int **smi\_ospf\_set\_if\_rtr\_dead\_interval\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifDeadInterval, u\_int32\_t vrId)
- int **smi\_ospf\_set\_if\_poll\_interval\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifPollInterval, u\_int32\_t vrId)
- int **smi\_ospf\_set\_if\_auth\_key\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int authKeyLength, char \*ifAuthKey, u\_int32\_t vrId)
- int **smi\_ospf\_set\_if\_status\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int areaStatus, u\_int32\_t vrId)
- int **smi\_ospf\_set\_if\_auth\_type\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifAuthType, u\_int32\_t vrId)
- int **smi\_ospf\_set\_if\_metric\_value\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int tos, int ifMetric, u\_int32\_t vrId)
- int **smi\_ospf\_set\_if\_metric\_status\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int tos, int ifMetricStatus, u\_int32\_t vrId)
- int **smi\_ospf\_set\_virt\_if\_retrans\_interval\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifRetransmitInterval, u\_int32\_t vrId)
- int **smi\_ospf\_set\_virt\_if\_status\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifStatus, u\_int32\_t vrId)
- int **smi\_ospf\_set\_nbr\_priority\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr, int ifIndex, int neighborPriority, u\_int32\_t vrId)
- int **smi\_ospf\_set\_nbma\_nbr\_status\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr, int ifIndex, int tableMgmtStatus, u\_int32\_t vrId)
- int **smi\_ospf\_set\_area\_aggregate\_status\_wrapper\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, char \*areaId, int addrAggrType, struct pal\_in4\_addr ipAddr, struct pal\_in4\_addr mask, int status, u\_int32\_t vrId)
- int **smi\_ospf\_set\_area\_aggregate\_status\_wrapper** (struct smiclient\_globals \*azg, int ospfProcessId, char \*areaId, int addrAggrType, struct pal\_in4\_addr ipAddr, struct pal\_in4\_addr mask, int status, u\_int32\_t vrId)

- int **smi\_ospf\_set\_area\_aggregate\_effect\_wrapper\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, char \*areaId, int addrAggrType, struct pal\_in4\_addr ipAddr, struct pal\_in4\_addr mask, int areaAggrEffect, u\_int32\_t vrId)
- int **smi\_ospf\_set\_area\_aggregate\_effect\_wrapper** (struct smiclient\_globals \*azg, int ospfProcessId, char \*areaId, int addrAggrType, struct pal\_in4\_addr ipAddr, struct pal\_in4\_addr mask, int areaAggrEffect, u\_int32\_t vrId)
- int **smi\_ospf\_if\_passive\_interface\_unset\_validate** (struct smiclient\_globals \*azg, int vrId, char \*ifName)
- int **smi\_ospf\_graceful\_restart\_planned\_set\_sdkapi\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId)
- int **smi\_ospf\_graceful\_restart\_planned\_set\_sdkapi\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, bool\_t gracefulRestartPlanned)
  
- int **smi\_ospf\_graceful\_restart\_planned\_set\_sdkapi\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, bool\_t gracefulRestartPlanned)
- int **smi\_ospf\_area\_default\_cost\_value\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_int32\_t areaCost)
- int **smi\_ospf\_graceful\_restart\_set\_sdkapi\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int restartSeconds)
- int **smi\_ospf\_hitless\_restart\_set\_sdkapi\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int hitlessRestartSeconds)
- int **smi\_ospf\_if\_ip\_router\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr areaId, int areaFormat, int ospfProcessId, int ospfInterfaceInstanceId, char \*opt\_str)
- int **smi\_ospf\_log\_adj\_changes\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*logAdjacencyDebugType)
- int **smi\_ospf\_if\_ip\_router\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName, struct pal\_in4\_addr areaId, int areaFormat, int ospfProcessId, int ospfInterfaceInstanceId)
- int **smi\_ospf\_max\_area\_limit\_set\_sdkapi\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, u\_int32\_t vrId, u\_int32\_t maxAreaLimit)
- int **smi\_ospf\_log\_adj\_changes\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, char \*logAdjacencyDebugType)
- int **smi\_ospf\_if\_passive\_interface\_set\_validate** (struct smiclient\_globals \*azg, int vrId, char \*ifName)
- int **smi\_ospf\_set\_settrap\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, int trap\_flag\_bit, u\_int32\_t vrId)
- int **smi\_ospf\_timers\_spf\_validate\_and\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t spfStartDelay, u\_int32\_t spfMinDelay, u\_int32\_t spfMaxDelay)
- int **smi\_ospf\_graceful\_restart\_unset\_sdkapi\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId)
- int **smi\_ospf\_hitless\_restart\_unset\_sdkapi\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId)
- int **smi\_ospf\_process\_shut\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospfProcessId)

- int **smi\_ospf\_process\_shut\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospfProcessId)
- int **smi\_ospf\_process\_shut\_set\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospfProcessId, bool\_t ospfShutDown)
- int **smi\_ospf\_process\_shut\_set\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vrId, u\_int32\_t ospfProcessId, bool\_t ospfShutDown)
- int **smi\_ospf\_vlink\_retransmit\_interval\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int retransmitInterval)
- int **smi\_ospf\_graceful\_restart\_planned\_unset\_sdkapi\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId)
- int **smi\_ospf\_auto\_cost\_reference\_bandwidth\_type\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, int referenceBandwidth, char \*bandwidthType)
- int **smi\_ospf\_area\_auth\_by\_type\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, u\_char authType)
- int **smi\_ospf\_area\_nssa\_default\_orinate\_route\_map\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, char \*routeMapName)
- int **smi\_ospf\_set\_lsdb\_limit\_sdkapi\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, u\_int32\_t vrId, u\_int32\_t limit, int actionType, int isLsdbLimit)
- int **smi\_ospf\_set\_area\_aggregate\_route\_tag\_wrapper\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, char \*areaId, int areaLsdbType, struct pal\_in4\_addr prefix, struct pal\_in4\_addr mask, u\_int32\_t routeTag, u\_int32\_t vrId)
- int **smi\_ospf\_set\_area\_aggregate\_route\_tag\_wrapper** (struct smiclient\_globals \*azg, int ospfProcessId, char \*areaId, int areaLsdbType, struct pal\_in4\_addr prefix, struct pal\_in4\_addr mask, u\_int32\_t routeTag, u\_int32\_t vrId)
- int **smi\_ospf\_if\_cost\_value\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*name, u\_int32\_t cost)
- int **smi\_ospf\_summary\_address\_not\_advertise\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr summaryAddr, u\_char summaryMask)
- int **smi\_ospf\_max\_area\_limit\_unset\_sdkapi\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, u\_int32\_t vrId)
- int **smi\_ospf\_set\_if\_type\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr ipAddr, int ifIndex, int ifType, u\_int32\_t vrId)
- int **smi\_ospf\_debug\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int IsDebug)
- int **smi\_ospf\_no\_debug\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int IsDebug)
- int **smi\_debug\_ospf\_packet\_validate** (struct smiclient\_globals \*azg, int vrId, int packetType, int debugMode, int debugDetail)
- int **smi\_debug\_no\_ospf\_packet\_validate** (struct smiclient\_globals \*azg, int vrId, int packetType, int debugMode, int debugDetail)
- int **smi\_cspf\_default\_retry\_interval\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int16\_t retryInterval)

- int **smi\_cspf\_retry\_interval\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, u\_int16\_t retryInterval)
- int **smi\_ospf\_set\_import\_as\_extern\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, int areaType)
- int **smi\_ospf\_set\_virt\_if\_transit\_delay\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtTransmitDelay, u\_int32\_t vrId)
- int **smi\_ospf\_set\_virt\_if\_hello\_interval\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtHelloInterval, u\_int32\_t vrId)
- int **smi\_ospf\_set\_virt\_if\_rtr\_dead\_interval\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtDeadInterval, u\_int32\_t vrId)
- int **smi\_ospf\_set\_virt\_if\_auth\_type\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, int ifVirtAuthType, u\_int32\_t vrId)
- int **smi\_ospf\_set\_virt\_if\_auth\_key\_validate** (struct smiclient\_globals \*azg, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId, char \*ifVirtAuthKey, u\_int32\_t vrId)
- int **smi\_ospf\_restart\_graceful\_sdkapi\_validate** (struct smiclient\_globals \*azg, int restartSeconds, int restartReason, u\_int32\_t vrId)
- int **smi\_ospf\_set\_area\_aggregate\_route\_tag\_validate** (struct smiclient\_globals \*azg, int proc\_id, struct pal\_in4\_addr area\_id, int area\_lsd\_type, struct pal\_in4\_addr prefix, struct pal\_in4\_addr mask, u\_int32\_t route\_tag, u\_int32\_t vr\_id)
- int **smi\_ospf\_process\_clear\_all** (struct smiclient\_globals \*azg, u\_int32\_t vrId)
- int **smi\_ospf\_network\_unset\_sdkapi\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, int proc\_id, char \*addr)
- int **smi\_ospf\_network\_unset\_sdkapi\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, int proc\_id, char \*addr)
- int **smi\_ospf\_area\_set\_wrapper** (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, int proc\_id, char \*area\_id)
- int **smi\_ospf\_area\_set\_wrapper\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, int proc\_id, char \*area\_id)

### 2.1.1 Detailed Description

Provides API for managing OSPF. Open Shortest Path First (OSPF) is an interior gateway protocol that was designed for TCP/IP networks to address the scaling issues of distance-vector routing protocols such as RIP. The API provided in this file forms the basis of ZebOS OSPF management. These APIs are used by various north bound management interfaces like CLI, SNMP and SMI



## 2.1.2 Function Documentation

### 2.1.2.1 `int smi_debug_ospf_packet_set (struct smiclient_globals * azg, int vrId, int packetType, int debugMode, int debugDetail)`

Use this function to specify the packet debugging options for OSPF ZebOS information. `smi_debug_ospf_packet_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *packet\_type* Pass packet type as following:
  - SMI\_OSPF\_DBG\_PACKET\_ALL - Debug all type of OSPF packets
  - SMI\_OSPF\_DBG\_PACKET\_HELLO - Debug OSPF hello packets
  - SMI\_OSPF\_DBG\_PACKET\_DESC - Debug OSPF database description
  - SMI\_OSPF\_DBG\_PACKET\_REQ - Debug OSPF link state request
  - SMI\_OSPF\_DBG\_PACKET\_UPD - Debug OSPF link state update
  - SMI\_OSPF\_DBG\_PACKET\_ACK - Debug OSPF link state acknowledgment
- ← *debug\_mode* Pass debug mode as following:
  - SMI\_OSPF\_DBG\_PACKET - Debug both sent and received packets
  - SMI\_OSPF\_DBG\_PACKET\_SEND - Debug sent packets
  - SMI\_OSPF\_DBG\_PACKET\_RCV - Debug received packets
- ← *is\_detail* 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`

### 2.1.2.2 `int smi_debug_ospf_packet_unset (struct smiclient_globals * azg, int vrId, int packetType, int debugMode, int debugDetail)`

Use this function to disable the packet debugging options for OSPF ZebOS information. `smi_debug_ospf_packet_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID

← ***packet\_type*** Pass packet type as following:

SMI\_OSPF\_DBG\_PACKET\_ALL - Debug all type of OSPF packets  
 SMI\_OSPF\_DBG\_PACKET\_HELLO - Debug OSPF hello packets  
 SMI\_OSPF\_DBG\_PACKET\_DESC - Debug OSPF database description  
 SMI\_OSPF\_DBG\_PACKET\_REQ - Debug OSPF link state request  
 SMI\_OSPF\_DBG\_PACKET\_UPD - Debug OSPF link state update  
 SMI\_OSPF\_DBG\_PACKET\_ACK - Debug OSPF link state acknowledgment

← ***debug\_mode*** Pass debug mode as following:

SMI\_OSPF\_DBG\_PACKET - Debug both sent and recieved packets  
 SMI\_OSPF\_DBG\_PACKET\_SEND - Debug sent packets  
 SMI\_OSPF\_DBG\_PACKET\_RCV - Debug recieved packets

← ***is\_detail*** 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

#### 2.1.2.3 int smi\_ospf\_abr\_type\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId, u\_char areaBorderType)

Sets This function sets the OSPF area border route (ABR) type. smi\_ospf\_abr\_type\_set

#### Parameters:

← ***azg*** Pointer to the SMI client global structure  
 ← ***vrId*** Virtual Router Id  
 ← ***ospfProcessId*** OSPF process ID <0-65535>  
 ← ***areaBorderType*** The type of area border router:

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_ABR\_TYPE\_INVALID

### 2.1.2.4 int smi\_ospf\_abr\_type\_unset (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*)

Sets This function resets the configured ABR type. smi\_ospf\_abr\_type\_unset

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_ABR\_TYPE\_INVALID

### 2.1.2.5 int smi\_ospf\_area\_auth\_by\_type\_unset (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*, struct pal\_in4\_addr *areaId*, u\_char *authType*)

This function disables the authentication type for the area. smi\_ospf\_area\_auth\_by\_type\_unset

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *area\_id* OSPF area ID
- ← *type* The authentication type of area border router (OSPF\_AUTH\_NULL | OSPF\_AUTH\_SIMPLE | OSPF\_AUTH\_CRYPTOGRAPHIC)

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AUTH\_TYPE\_INVALID
- OSPF\_API\_SET\_ERR\_AUTH\_TYPE\_MISMATCH
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT

**2.1.2.6** `int smi_ospf_area_auth_type_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, u_char areaAuthType)`

This function enables authentication for an area. `smi_ospf_area_auth_type_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *areaAuthType* The authentication type of area border router (OSPF\_AUTH\_NULL | OSPF\_AUTH\_SIMPLE | OSPF\_AUTH\_CRYPTOGRAPHIC)

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AUTH\_TYPE\_INVALID
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT

**2.1.2.7** `int smi_ospf_area_auth_type_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId)`

This function disables the authentication type for the area. `smi_ospf_area_auth_type_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AUTH\_TYPE\_INVALID
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT

**2.1.2.8** `int smi_ospf_area_default_cost_set (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId,  
u_int32_t areaCost)`

This function assigns the specified cost to the default summary route used for a not so stubby area (NSSA). `smi_ospf_area_default_cost_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *areaCost* The default cost for the area <0-16777215>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_COST\_INVALID
- OSPF\_API\_SET\_ERR\_AREA\_IS\_DEFAULT
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT

**2.1.2.9** `int smi_ospf_area_default_cost_unset (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId)`

This function resets the cost to the default value of 1. `smi_ospf_area_default_cost_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_DEFAULT

### 2.1.2.10 `int smi_ospf_area_default_cost_value_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, u_int32_t areaCost)`

This function assigns the specified cost to the default summary route used for a not so stubby area (NSSA). `smi_ospf_area_default_cost_value_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *area\_id* OSPF area ID
- ← *cost* The default cost for the area <0-16777215>

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_COST\_INVALID
- OSPF\_API\_SET\_ERR\_AREA\_IS\_DEFAULT
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT

### 2.1.2.11 `int smi_ospf_area_export_list_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, char * accessListName)`

This function sets the type-3 export filter for networks announced to other areas. `smi_ospf_area_export_list_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *accessListName* The name of the access list.

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.12 `int smi_ospf_area_export_list_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId)`

This function resets the export list. `smi_ospf_area_export_list_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_LIMIT  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

### 2.1.2.13 `int smi_ospf_area_filter_list_access_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char * areaId, int accessType, char * accessName)`

This function sets to filter prefixes advertise in type-3 link-state advertisement (LSAs) with the access list name between OSPF areas of an ABR. `smi_ospf_area_filter_list_access_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *areaFilterType* Filter type ( FILTER\_IN | FILTER\_OUT)
- ← *prefixListName* Name of the prefix list.

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_LIMIT  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.14 `int smi_ospf_area_filter_list_access_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char * areaId, int accessType)`

This function resets the filter list access configuration to either FILTER\_IN or FILTER\_OUT. smi\_ospf\_area\_filter\_list\_access\_unset

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *areaFilterType* Filter type ( FILTER\_IN | FILTER\_OUT)

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

### 2.1.2.15 `int smi_ospf_area_filter_list_prefix_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char * areaId, int prefixType, char * prefixName)`

This function sets the type of filter prefix advertised in type-3 LSAs between the OSPF areas of an ABR. smi\_ospf\_area\_filter\_list\_prefix\_set

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *areaFilterType* Filter type ( FILTER\_IN | FILTER\_OUT)
- ← *prefixListName* Name of the prefix list.

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_INVALID\_FILTER\_TYPE  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_AREA\_LIMIT



### 2.1.2.16 `int smi_ospf_area_filter_list_prefix_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char * areaId, int prefixType)`

This function cancels the filter prefix advertise. `smi_ospf_area_filter_list_prefix_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *areaFilterType* Filter type ( FILTER\_IN | FILTER\_OUT)

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

### 2.1.2.17 `int smi_ospf_area_import_list_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, char * accessListName)`

This function sets the import list value for the type-3 import filter. `smi_ospf_area_import_list_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *accessListName* The name of the list.

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.18 `int smi_ospf_area_import_list_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId)`

This function resets the import list value for the type-3 import. `smi_ospf_area_import_list_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

### 2.1.2.19 `int smi_ospf_area_no_summary_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId)`

This function sets the OSPF area as stub. `smi_ospf_area_no_summary_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_DEFAULT
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST

### 2.1.2.20 `int smi_ospf_area_no_summary_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId)`

This function disables the OSPF area as stub. `smi_ospf_area_no_summary_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_DEFAULT
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST

**2.1.2.21** `int smi_ospf_area_nssa_default_originate_metric_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, int nssaDefaultOriginMetric)`

Sets the default metric for the routes originated from this NSSA router. `smi_ospf_area_nssa_default_originate_metric_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *nssaDefaultOriginMetric* Metric value <0-16777214>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_NSSA
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_METRIC\_INVALID

### 2.1.2.22 `int smi_ospf_area_nssa_default_originate_metric_type_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, int nssaDefaultOriginMetricType)`

Sets the default metric type for the routes originated from this NSSA router. `smi_ospf_area_nssa_default_originate_metric_type_set`

#### Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrId** Virtual Router Id
- ← **ospfProcessId** OSPF process ID <0-65535>
- ← **areaId** OSPF area ID
- ← **nssaDefaultOriginMetricType** Metric type <1-2>  
External Type1|Type2

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_NSSA
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_METRIC\_TYPE\_INVALID

### 2.1.2.23 `int smi_ospf_area_nssa_default_originate_route_map_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, char * routeMapName)`

Sets the route-map for the routes originated from this NSSA router. `smi_ospf_area_nssa_default_originate_route_map_set`

#### Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vrId** Virtual Router Id
- ← **ospfProcessId** OSPF process ID <0-65535>
- ← **areaId** OSPF area ID
- ← **routeMapName** Name of the Route-map

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

```

OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_ERR_AREA_NOT_NSSA
OSPF_API_SET_ERR_PROCESS_NOT_EXIST
OSPF_API_SET_ERR_AREA_IS_BACKBONE
OSPF_API_SET_ERR_AREA_NOT_EXIST
OSPF_API_SET_ERR_PROCESS_ID_INVALID

```

#### 2.1.2.24 int smi\_ospf\_area\_nssa\_default\_originate\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

Sets the originate Type-7 default LSA into the NSSA. smi\_ospf\_area\_nssa\_default\_originate\_set

##### Parameters:

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *area\_id* OSPF area ID

##### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_NSSA  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

#### 2.1.2.25 int smi\_ospf\_area\_nssa\_default\_originate\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

Unsets the originate Type-7 default LSA into the NSSA. smi\_ospf\_area\_nssa\_default\_originate\_unset

##### Parameters:

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_NSSA
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

### 2.1.2.26 **int smi\_ospf\_area\_nssa\_no\_redistribution\_set** (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*, struct pal\_in4\_addr *areaId*)

This function sets OSPF redistribution in not allowed to the stub. smi\_ospf\_area\_nssa\_no\_redistribution\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_NSSA
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

### 2.1.2.27 **int smi\_ospf\_area\_nssa\_no\_redistribution\_unset** (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*, struct pal\_in4\_addr *areaId*)

This function resets OSPF: redistribution in not allowed to the stub. smi\_ospf\_area\_nssa\_no\_redistribution\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_NSSA
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

### 2.1.2.28 int smi\_ospf\_area\_nssa\_set (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*, struct pal\_in4\_addr *areaId*)

This function defines an area as Not So Stubby Area. smi\_ospf\_area\_nssa\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT
- OSPF\_API\_SET\_ERR\_AREA\_HAS\_VLINK
- OSPF\_API\_SET\_ERR\_AREA\_IS\_STUB

### 2.1.2.29 int smi\_ospf\_area\_nssa\_stability\_interval\_set (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*, struct pal\_in4\_addr *areaId*, u\_int32\_t *nssaStabilityInterval*)

This function sets the NSSA stability interval. smi\_ospf\_area\_nssa\_stability\_interval\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

← *nssaStabilityInterval* NSSA stability interval in seconds <0-2147483647>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_NSSA  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

**2.1.2.30** `int smi_ospf_area_nssa_translator_role_set (struct smiclient_globals * azg, u_int32_t vrid, int ospfProcessId, struct pal_in4_addr areaId, u_char nssaTranslatorRole)`

This function sets the Area border router to be the translator between the types. This is applied only if the area type is NSSA. `smi_ospf_area_nssa_translator_role_set`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrid* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* OSPF area ID  
 ← *nssaTranslatorRole* NSSA-ABR translator role(OSPF\_NSSA\_TRANSLATE\_CANDIDATE | OSPF\_NSSA\_TRANSLATE\_ALWAYS)

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_NSSA  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST

**2.1.2.31** `int smi_ospf_area_nssa_translator_role_unset (struct smiclient_globals * azg, u_int32_t vrid, int ospfProcessId, struct pal_in4_addr areaId)`

Removes the value of the translator parameter of the area. `smi_ospf_area_nssa_translator_role_unset`

**Parameters:**

← *azg* Pointer to the SMI client global structure



← *vrId* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_NSSA  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

### 2.1.2.32 int smi\_ospf\_area\_nssa\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId)

This function removes the NSSA designation from the specified area. smi\_ospf\_area\_nssa\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST

### 2.1.2.33 int smi\_ospf\_area\_range\_not\_advertise\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr ospfAreaRange, u\_char rangeMask)

This function sets the ABR to not advertise the summary LSA for each route in a specific adjacent area. smi\_ospf\_area\_range\_not\_advertise\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure

- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *ospfAreaRange* The network address range.
- ← *rangeMask* The mask length from the /M sub-parameter of the net parameter

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_AREA\_RANGE\_NOT\_EXIST

**2.1.2.34** `int smi_ospf_area_range_not_advertise_unset (struct smiclient_globals *azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr ospfAreaRange, u_char rangeMask)`

This function allows the area border router (ABR) to create a summary LSA for each route in specific area and advertise it in adjacent areas. `smi_ospf_area_range_not_advertise_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *ospfAreaRange* The network address range.
- ← *rangeMask* The mask length from the /M sub-parameter of the net parameter

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_AREA\_RANGE\_NOT\_EXIST

**2.1.2.35** `int smi_ospf_area_range_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr ospfAreaRange, u_char rangeMask)`

This function specifies an address range, for which to advertise a single route to other areas by the ABRs. `smi_ospf_area_range_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *ospfAreaRange* The network address range.
- ← *rangeMask* The mask length from the /M sub-parameter of the net parameter

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_LIMIT  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_AREA\_IS\_DEFAULT  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.36** `int smi_ospf_area_range_substitute_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char * areaId, char * ospfAreaRange, char * substituteAddr)`

This function summarizes routes via Matching addresses and or masks, border routers only and Announcing area range as a prefix. `smi_ospf_area_range_substitute_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *ospfAreaRange* The network address range.
- ← *rangeMask* The mask length from the /M sub-parameter of the net parameter
- ← *substituteNet* The substitute network.
- ← *substituteMasklen* The substitute prefix length.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.37 `int smi_ospf_area_range_substitute_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char * areaId, char * ospfAreaRange)`

This function cancels the routes via Matching addresses and or masks, border routers only and Announcing area range as a prefix. `smi_ospf_area_range_substitute_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *ospfAreaRange* The network address range.
- ← *rangeMask* The mask length from the /M sub-parameter of the net parameter

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_RANGE\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST

### 2.1.2.38 `int smi_ospf_area_range_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr ospfAreaRange, u_char rangeMask)`

This function deletes the area range. `smi_ospf_area_range_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id

- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *ospfAreaRange* The network address range.
- ← *rangeMask* The mask length from the /M sub-parameter of the net parameter

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_AREA\_IS\_DEFAULT
- OSPF\_API\_SET\_ERR\_MALLOC\_ERR

**2.1.2.39** `int smi_ospf_area_shortcut_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, u_char areaShortcutType)`

This function sets the shortcut mode of the specified area. Area shortcut enables traffic to go through the nonbackbone area with a lower metric, whether or not the ABR is attached to the backbone area. `smi_ospf_area_shortcut_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *areaShortcutType* The type of shortcut ( 0 - default | 1 - enable | 2 - disable )

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_INVALID\_VALUE

**2.1.2.40** `int smi_ospf_area_shortcut_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId)`

This function removes the shortcut mode of the specified area. `smi_ospf_area_shortcut_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE

#### 2.1.2.41 int smi\_ospf\_area\_stub\_set (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*, struct pal\_in4\_addr *areaId*)

This function sets the specified area as a stub area. smi\_ospf\_area\_stub\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT
- OSPF\_API\_SET\_ERR\_AREA\_HAS\_VLINK
- OSPF\_API\_SET\_ERR\_AREA\_IS\_NSSA

#### 2.1.2.42 int smi\_ospf\_area\_stub\_unset (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*, struct pal\_in4\_addr *areaId*)

This function removes the stub definition from the specified area. smi\_ospf\_area\_stub\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_IS\_NSSA

#### 2.1.2.43 `int smi_ospf_auto_cost_reference_bandwidth_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, int referenceBandwidth)`

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

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *referenceBandwidth* The bandwidth in Mbits/second <1-4294967>.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_REFERENCE\_BANDWIDTH\_INVALID

#### 2.1.2.44 `int smi_ospf_auto_cost_reference_bandwidth_type_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, int referenceBandwidth, char * bandwidthType)`

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

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *referenceBandwidth* The bandwidth in Mbits/second <1-4294967>.
- In*] bandwidth String pointer for gbps or mpbs option.

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_REFERENCE\_BANDWIDTH\_INVALID

#### 2.1.2.45 int smi\_ospf\_auto\_cost\_reference\_bandwidth\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId)

Sets This function sets the reference bandwidth value. OSPF calculates the OSPF metric for an interface by dividing the reference bandwidth. smi\_ospf\_auto\_cost\_reference\_bandwidth\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

#### 2.1.2.46 int smi\_ospf\_capability\_cspf\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId)

This function sets the Constrained Shortest Path First (CSPF) capability for an OSPF process. The CSPF protocol module relies on the OSPF database to calculate the shortest path through the network. smi\_ospf\_capability\_cspf\_set

**Parameters:**

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



← *vrId* Virtual Router Id  
← *ospfProcessId* OSPF process ID <0-65535>.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_CSPF\_INSTANCE\_EXIST

**2.1.2.47 int smi\_ospf\_capability\_cspf\_unset (struct smiclient\_globals \* *azg*,  
u\_int32\_t *vrId*, int *ospfProcessId*)**

This function disables the CSPF capability for an OSPF process. smi\_ospf\_capability\_cspf\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *vrId* Virtual Router Id  
← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_CSPF\_INSTANCE\_EXIST

**2.1.2.48 int smi\_ospf\_capability\_opaque\_lsa\_set (struct smiclient\_globals \* *azg*,  
u\_int32\_t *vrId*, int *ospfProcessId*)**

This function sets the opaque capability for an OSPF process. smi\_ospf\_capability\_opaque\_lsa\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *vrId* Virtual Router Id  
← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.49 int smi\_ospf\_capability\_opaque\_lsa\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId)**

This function unsets the opaque capability for an OSPF process. smi\_ospf\_capability\_opaque\_lsa\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.50 int smi\_ospf\_capability\_restart\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId, int restartMethod)**

Enables the OSPF restart capability by graceful restart or signalling restart method. smi\_ospf\_capability\_restart\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router ID  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *restartMethod* Garaceful or signalling restart method

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_RESTART\_METHOD\_INVALID

**2.1.2.51** `int smi_ospf_capability_restart_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

Disables the OSPF restart capability. smi\_ospf\_capability\_restart\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.52** `int smi_ospf_capability_traffic_engineering_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

This function sets the traffic engineering capability for an OSPF process. smi\_ospf\_capability\_traffic\_engineering\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.53** `int smi_ospf_capability_traffic_engineering_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

This function resets the traffic engineering capability for an OSPF process. smi\_ospf\_capability\_traffic\_engineering\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id

← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.54 int smi\_ospf\_compatible\_rfc1583\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId)**

Sets This function calculates route summary costs according to RFC 1583. smi\_ospf\_compatible\_rfc1583\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_ABR\_TYPE\_INVALID

**2.1.2.55 int smi\_ospf\_compatible\_rfc1583\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId)**

Sets This function disables the calculation of route summary costs according to RFC 1583. smi\_ospf\_compatible\_rfc1583\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

```

OSPF_API_SET_ERR_PROCESS_ID_INVALID
OSPF_API_SET_ERR_PROCESS_NOT_EXIST
OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_ERR_ABR_TYPE_INVALID

```

### 2.1.2.56 int smi\_ospf\_cspf\_better\_protection\_type (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId, bool\_t cspfProtectionType)

This function enables/disables the default cspf protection type. smi\_ospf\_cspf\_better\_protection\_type

#### Parameters:

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>.  
 ← *cspfProtectionType*

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_CSPF\_INSTANCE\_NOT\_EXIST

### 2.1.2.57 int smi\_ospf\_debug\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, int debug)

Use this function to specify debugging option for OSPF ZebOS information. smi\_ospf\_debug\_set

#### Parameters:

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router ID  
 ← *debug* Pass debug flag as following:  
     SMI\_OSPF\_DBG\_ALL - Enable or disable debugging for ifsm,nsfm,lsa,nsm,events and route  
     SMI\_OSPF\_DBG\_IFSM - Debug OSPF Interface State Machine  
     SMI\_OSPF\_DBG\_IFSM\_EVENTS - Debug OSPF Interface State Machine events information  
     SMI\_OSPF\_DBG\_IFSM\_STATUS - Debug OSPF Interface State Machine status information

SMI\_OSPF\_DBG\_IFSM\_TIMERS - Debug OSPF Interface State Machine timers information

SMI\_OSPF\_DBG\_NFSM - Debug OSPF Neighbor State Machine

SMI\_OSPF\_DBG\_NFSM\_EVENTS - Debug OSPF Neighbor State Machine events information

SMI\_OSPF\_DBG\_NFSM\_STATUS - Debug OSPF Neighbor State Machine status information

SMI\_OSPF\_DBG\_NFSM\_TIMERS - Debug OSPF Neighbor State Machine timers information

SMI\_OSPF\_DBG\_LSA - Debug OSPF Link State Advertisement

SMI\_OSPF\_DBG\_LSA\_FLOODING - Debug LSA flooding

SMI\_OSPF\_DBG\_LSA\_GENERATE - Debug LSA generation

SMI\_OSPF\_DBG\_LSA\_INSTALL - Debug LSA installation

SMI\_OSPF\_DBG\_LSA\_MAXAGE - Debug the maximum age processing

SMI\_OSPF\_DBG\_LSA\_REFRESH - Debug LSA refresh

SMI\_OSPF\_DBG\_NSM - Debug OSPF NSM information

SMI\_OSPF\_DBG\_NSM\_INTERFACE - Debug NSM interface information

SMI\_OSPF\_DBG\_NSM\_REDISTRIBUTE - Debug NSM redistribute information

SMI\_OSPF\_DBG\_RIB - Debug OSPF RIB information

SMI\_OSPF\_DBG\_RIB\_INTERFACE - Debug RIB interface information

SMI\_OSPF\_DBG\_RIB\_REDISTRIBUTE - Debug RIB redistribute information

SMI\_OSPF\_DBG\_EVENT - Debug OSPF event trouble shooting

SMI\_OSPF\_DBG\_EVENT\_ABR - Debug OSPF ABR events

SMI\_OSPF\_DBG\_EVENT\_ASBR - Debug OSPF ASBR events

SMI\_OSPF\_DBG\_EVENT\_LSA - Debug OSPF LSA events

SMI\_OSPF\_DBG\_EVENT\_NSSA - Debug OSPF NSSA events

SMI\_OSPF\_DBG\_EVENT\_OS - Debug OSPF OS events

SMI\_OSPF\_DBG\_EVENT\_ROUTER - Debug OSPF ROUTER events

SMI\_OSPF\_DBG\_EVENT\_VLINK - Debug OSPF VLINK events

SMI\_OSPF\_DBG\_ROUTE\_CALC - Debug route calculation

SMI\_OSPF\_DBG\_ROUTE\_CALC\_ASE - Debug OSPF external route calculation

SMI\_OSPF6\_DBG\_ROUTE\_CALC\_IA - Debug OSPF inter area calculation

SMI\_OSPF\_DBG\_ROUTE\_INSTALL - Debug OSPF route installation

SMI\_OSPF\_DBG\_ROUTE\_CALC\_SPF - Debug OSPF SPF calculation

SMI\_OSPF\_DBG\_BFD - Debug Bidirectional Forwarding Detection

SMI\_OSPF\_DBG\_CSPF - Debug CSPF event information

SMI\_OSPF\_DBG\_CSPF\_HEXDUMP - Debug CSPF message hexdump

SMI\_OSPF\_DBG\_DB\_TIMER - Debug OSPF Database Timers

**Returns:**

0 on success, otherwise one of the following error codes OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
SMI\_ERROR

**2.1.2.58 int smi\_ospf\_debug\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, int debug)**

Use this function to disable debugging option for OSPF ZebOS information. smi\_ospf\_debug\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *debug* Pass debug flag as following:
  - SMI\_OSPF\_DBG\_ALL - Enable or disable debugging for ifsm,nsfm,lsa,nsm,events and route
  - SMI\_OSPF\_DBG\_IFSM - Debug OSPF Interface State Machine
  - SMI\_OSPF\_DBG\_IFSM\_EVENTS - Debug OSPF Interface State Machine events information
  - SMI\_OSPF\_DBG\_IFSM\_STATUS - Debug OSPF Interface State Machine status information
  - SMI\_OSPF\_DBG\_IFSM\_TIMERS - Debug OSPF Interface State Machine timers information
  - SMI\_OSPF\_DBG\_NFSM - Debug OSPF Neighbor State Machine
  - SMI\_OSPF\_DBG\_NFSM\_EVENTS - Debug OSPF Neighbor State Machine events information
  - SMI\_OSPF\_DBG\_NFSM\_STATUS - Debug OSPF Neighbor State Machine status information
  - SMI\_OSPF\_DBG\_NFSM\_TIMERS - Debug OSPF Neighbor State Machine timers information
  - SMI\_OSPF\_DBG\_LSA - Debug OSPF Link State Advertisement
  - SMI\_OSPF\_DBG\_LSA\_FLOODING - Debug LSA flooding
  - SMI\_OSPF\_DBG\_LSA\_GENERATE - Debug LSA generation
  - SMI\_OSPF\_DBG\_LSA\_INSTALL - Debug LSA installation
  - SMI\_OSPF\_DBG\_LSA\_MAXAGE - Debug the maximum age processing
  - SMI\_OSPF\_DBG\_LSA\_REFRESH - Debug LSA refresh
  - SMI\_OSPF\_DBG\_NSM - Debug OSPF NSM information
  - SMI\_OSPF\_DBG\_NSM\_INTERFACE - Debug NSM interface information
  - SMI\_OSPF\_DBG\_NSM\_REDISTRIBUTE - Debug NSM redistribute information
  - SMI\_OSPF\_DBG\_RIB - Debug OSPF RIB information
  - SMI\_OSPF\_DBG\_RIB\_INTERFACE - Debug RIB interface information

SMI\_OSPF\_DBG\_RIB\_REDISTRIBUTE - Debug RIB redistribute information  
 SMI\_OSPF\_DBG\_EVENT - Debug OSPF event trouble shooting  
 SMI\_OSPF\_DBG\_EVENT\_ABR - Debug OSPF ABR events  
 SMI\_OSPF\_DBG\_EVENT\_ASBR - Debug OSPF ASBR events  
 SMI\_OSPF\_DBG\_EVENT\_LSA - Debug OSPF LSA events  
 SMI\_OSPF\_DBG\_EVENT\_NSSA - Debug OSPF NSSA events  
 SMI\_OSPF\_DBG\_EVENT\_OS - Debug OSPF OS events  
 SMI\_OSPF\_DBG\_EVENT\_ROUTER - Debug OSPF ROUTER events  
 SMI\_OSPF\_DBG\_EVENT\_VLINK - Debug OSPF VLINK events  
 SMI\_OSPF\_DBG\_ROUTE\_CALC - Debug route calculation  
 SMI\_OSPF\_DBG\_ROUTE\_CALC\_ASE - Debug OSPF external route calculation  
 SMI\_OSPF6\_DBG\_ROUTE\_CALC\_IA - Debug OSPF inter area calculation  
 SMI\_OSPF\_DBG\_ROUTE\_INSTALL - Debug OSPF route installation  
 SMI\_OSPF\_DBG\_ROUTE\_CALC\_SPF - Debug OSPF SPF calculation  
 SMI\_OSPF\_DBG\_BFD - Debug Bidirectional Forwarding Detection  
 SMI\_OSPF\_DBG\_CSPF - Debug CSPF event information  
 SMI\_OSPF\_DBG\_CSPF\_HEXDUMP - Debug CSPF message hexdump  
 SMI\_OSPF\_DBG\_DB\_TIMER - Debug OSPF Database Timers

**Returns:**

0 on success, otherwise one of the following error codes OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 SMI\_ERROR

**2.1.2.59 int smi\_ospf\_default\_metric\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId, int metricValue)**

Sets the default metric value for OSPF to redistribute routes.

A default metric facilitates redistributing routes with incompatible metrics. smi\_ospf\_default\_metric\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *metricValue* Metric value <1-16777214>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST



OSPF\_API\_SET\_ERR\_METRIC\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

#### 2.1.2.60 `int smi_ospf_default_metric_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

Unsets the default metric value for OSPF to redistribute routes. `smi_ospf_default_metric_unset`

##### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>

##### Returns:

0 on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_METRIC\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

#### 2.1.2.61 `int smi_ospf_disable_db_summary_opt (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

This function disables the OSPF Database Summary List optimization. The `OSPF_DB_SUMMARY_OPT` flag is unset to indicate the feature is disabled. `smi_ospf_disable_db_summary_opt`

##### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

##### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.62 `int smi_ospf_disable_ext_multi_inst (struct smiclient_globals * azg, u_int32_t vrId)`

This call implements the no parameter of the enable ext-ospf-multi-inst command to disable support of multiple OSPF instances. `smi_ospf_disable_ext_multi_inst`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_EXT\_MULTI\_INST\_NOT\_ENABLED

### 2.1.2.63 `int smi_ospf_distance_all_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, int adminDistance)`

Sets the OSPF administrative distance.

The administrative distance rates the trustworthiness of a routing information source.

The higher distance value implies the lower trustworthiness. `smi_ospf_distance_all_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *adminDistance* OSPF administrative distance <1-255>

#### Returns:

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_DISTANCE\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

### 2.1.2.64 `int smi_ospf_distance_all_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, int adminDistance)`

Unsets the OSPF administrative distance. `smi_ospf_distance_all_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *adminDistance* Administrative distance <1-255>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

### 2.1.2.65 int smi\_ospf\_distance\_external\_set (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*, int *externalDistance*)

Sets the OSPF administrative distance for all routes learned from other routing domains learned via redistribution.

The administrative distance rates the trustworthiness of a routing information source.

The higher distance value implies the lower trustworthiness. smi\_ospf\_distance\_external\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *externalDistance* OSPF administrative distance <1-255>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_DISTANCE\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

### 2.1.2.66 int smi\_ospf\_distance\_external\_unset (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*)

Unsets the OSPF administrative distance for all routes learned from other routing domains learned via redistribution. smi\_ospf\_distance\_external\_unset

**Parameters:**

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

← *vrId* Virtual Router ID  
 ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

### 2.1.2.67 int smi\_ospf\_distance\_inter\_area\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId, int interAreaDistance)

Sets the OSPF administrative distance for all routes from one area to another area (i.e., inter-area).

The administrative distance rates the trustworthiness of a routing information source.

The higher distance value implies the lower trustworthiness. smi\_ospf\_distance\_inter\_area\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router ID  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *interAreaDistance* OSPF administrative distance <1-255>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_DISTANCE\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

### 2.1.2.68 int smi\_ospf\_distance\_inter\_area\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId)

Unsets the OSPF administrative distance for all routes from one area to another area (i.e., inter-area). smi\_ospf\_distance\_inter\_area\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router ID  
 ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.69 int smi\_ospf\_distance\_intra\_area\_set (struct smiclient\_globals \* *azg*,  
 u\_int32\_t *vrId*, int *ospfProcessId*, int *intraAreaDistance*)**

Sets the OSPF administrative distance for all routes within an area (i.e., intra-area).

The administrative distance rates the trustworthiness of a routing information source.

The higher distance value implies the lower trustworthiness. smi\_ospf\_distance\_intra\_area\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router ID  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *intraAreaDistance* OSPF administrative distance <1-255>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_DISTANCE\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.70 int smi\_ospf\_distance\_intra\_area\_unset (struct smiclient\_globals \* *azg*,  
 u\_int32\_t *vrId*, int *ospfProcessId*)**

Unsets the OSPF administrative distance for all routes within an area (i.e., intra-area).  
 smi\_ospf\_distance\_intra\_area\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router ID  
 ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.71** `int smi_ospf_distance_source_set (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, u_int32_t adminDistance, struct  
pal_in4_addr sourceIpAddr, u_char masklen, char * accessListName)`

Sets the OSPF administrative distance to prefixes whose nexthop matches the given source IP address.

The administrative distance rates the trustworthiness of a routing information source.

The higher distance value implies the lower trustworthiness. `smi_ospf_distance_source_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *adminDistance* OSPF administrative distance <1-255>
- ← *sourceIpAddr* IP source prefix
- ← *masklen* Length of mask
- ← *accessListName* Name of access list

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.72** `int smi_ospf_distance_source_unset (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, struct pal_in4_addr sourceIpAddr,  
u_char masklen, char * accessListName)`

Unsets the OSPF administrative distance to prefixes whose nexthop matches the given source IP address. `smi_ospf_distance_source_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *sourceIpAddr* IP source prefix
- ← *masklen* Length of mask
- ← *accessListName* Name of access list

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_DISTANCE\_NOT\_EXIST

**2.1.2.73 int smi\_ospf\_distribute\_list\_in\_set (struct smiclient\_globals \* azg,  
 u\_int32\_t vrId, int ospfProcessId, char \* accessListName)**

Sets OSPF to allow routing updates from particular access list into this OSPF instance (i.e filters networks from particular access list at ingress). smi\_ospf\_distribute\_list\_in\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router ID  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *accessListName* Access list name

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_MALLOC\_ERR

**2.1.2.74 int smi\_ospf\_distribute\_list\_in\_unset (struct smiclient\_globals \* azg,  
 u\_int32\_t vrId, int ospfProcessId, char \* accessListName)**

Unsets OSPF to allow routing updates from particular access list into this OSPF instance (i.e filters networks from particular access list at ingress). smi\_ospf\_distribute\_list\_in\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router ID  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *accessListName* Access list name

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.75** `int smi_ospf_distribute_list_out_set (struct smiclient_globals *  
azg, u_int32_t vrId, int ospfProcessId, int routeSourceType, int  
secondaryOspfProcessId, char * accessListName)`

Sets OSPF to distribute routing updates from particular routing protocol into another OSPF instance (i.e filters networks from particular outgoing routing updates). `smi_ospf_distribute_list_out_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceType* Source for the route (1|2|3|4|6|8|9)  
1-Kernel, 2-Connected, 3-Static,  
4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *secondaryOspfProcessId* Another OSPF instance ID <1-65535>
- ← *accessListName* Access list name

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.76** `int smi_ospf_distribute_list_out_unset (struct smiclient_globals  
* azg, u_int32_t vrId, int ospfProcessId, int routeSourceType, int  
secondaryOspfProcessId, char * accessListName)`

Unsets OSPF from distributing routing updates from particular routing protocol into another OSPF instance (i.e filters networks from particular outgoing routing updates). `smi_ospf_distribute_list_out_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourcetype* Source for the route (1|2|3|4|6|8|9)  
1-Kernel, 2-Connected, 3-Static,  
4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *secondaryOspfProcessId* Another OSPF instance ID <1-65535>
- ← *accessListName* Access list name



**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.77 int smi\_ospf\_dna\_set\_sdkapi (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*)**

Set DNA. smi\_ospf\_dna\_set\_sdkapi

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router ID  
 ← *ospfProcessId* Process ID return 0 on success, otherwise one of the following error code

**2.1.2.78 int smi\_ospf\_dna\_unset\_sdkapi (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*)**

Unset DNA. smi\_ospf\_dna\_unset\_sdkapi

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router ID  
 ← *ospfProcessId* Process ID return 0 on success, otherwise one of the following error code

**2.1.2.79 int smi\_ospf\_domain\_id\_set (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*, char \* *domainType*, int *domainValue*, bool\_t *isPrimaryDomainId*)**

Sets This function sets an OSPF domain ID as specified: domain ID for a particular OSPF VRF instance. smi\_ospf\_domain\_id\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *domainType*

- ← *domainValue* Domain ID value entered through the CLI.
- ← *isPrimaryDomainId* Boolean flag to identify whether or not the entered domain ID is primary.

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_INVALID\_HEX\_VALUE

**2.1.2.80** `int smi_ospf_domain_id_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char * domainType, u_int8_t * domainValue, bool_t isPrimaryDomainId)`

Sets This function removes the OSPF domain ID. smi\_ospf\_domain\_id\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *domainType* Domain ID type
- ← *domainValue* Domain ID value entered through the CLI.
- ← *isPrimaryDomainId* Boolean flag to identify whether or not the entered domain ID is primary.

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_INVALID\_HEX\_VALUE

**2.1.2.81** `int smi_ospf_enable_db_summary_opt (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

This function enables the OSPF Database Summary List optimization. The OSPF\_DB\_SUMMARY\_OPT flag is set to indicate the feature is enabled. smi\_ospf\_enable\_db\_summary\_opt

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.82 int smi\_ospf\_enable\_ext\_multi\_inst (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*)

Sets This function enables multiple OSPF instances to run on a subnet. smi\_ospf\_enable\_ext\_multi\_inst

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.83 int smi\_ospf\_get\_address\_less\_if (struct smiclient\_globals \* *azg*, int *ospfProcessId*, struct pal\_in4\_addr *addr*, int *ifIndex*, int \* *addrLessIf*, u\_int32\_t *vrId*)

Gets the ifIndex for the purpose of easing the instancing of addressed and address-less interfaces; this variable takes the value 0 on interfaces with IP addresses and the corresponding value of ifIndex for interfaces having no IP address. smi\_ospf\_get\_address\_less\_if

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero

→ *addrLessIf* Interface index or zero

← *vrId* Virtual Router ID

#### Returns:

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_GET\_ERROR

#### 2.1.2.84 int smi\_ospf\_get\_admin\_stat (struct smiclient\_globals \* azg, int ospfProcessId, int \* adminStatus, u\_int32\_t vrId)

Gets the administrative status of given OSPF instance in the router.

The value 'enabled' denotes that the OSPF Process is active on at least one interface; 'disabled' disables it on all interfaces. smi\_ospf\_get\_admin\_stat

#### Parameters:

← *azg* Pointer to the SMI client global structure

← *ospfProcessId* OSPF process ID <0-65535>

→ *status* Enabled/disabled

← *vrId* Virtual Router ID

#### Returns:

0 on success, otherwise one of the following error codes

#### 2.1.2.85 int smi\_ospf\_get\_area\_aggregate\_area\_id (struct smiclient\_globals \* azg, int ospfProcessId, struct pal\_in4\_addr areaId, int type, struct pal\_in4\_addr addr, struct pal\_in4\_addr mask, struct pal\_in4\_addr \* areaAggrAreaId, u\_int32\_t vrId)

Gets the area within which the address aggregate is to be found. smi\_ospf\_get\_area\_aggregate\_area\_id

#### Parameters:

← *azg* Pointer to the SMI client global structure

← *ospfProcessId* OSPF process ID <0-65535>.

← *area\_id* Aggregate Area ID

← *type* Address aggregate type.

← *addr* IP address of the Net or Subnet

← *mask* Subnet mask

→ *areaAggrAreaId* Aggregate Area ID

← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.86** `int smi_ospf_get_area_aggregate_effect (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int type, struct pal_in4_addr addr, struct pal_in4_addr mask, int * areaAggrEffect, u_int32_t vrId)`

Gets and object that tells, subnets subsumed by ranges either trigger the advertisement of the indicated aggregate (advertiseMatching) or result in the subnet's not being advertised at all outside the area. smi\_ospf\_get\_area\_aggregate\_effect

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>.  
 ← *area\_id* Aggregate Area ID  
 ← *type* Address aggregate type.  
 ← *addr* IP address of the Net or Subnet  
 ← *mask* Subnet mask  
 → *areaAggrEffect* Management object for subnets  
 ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.87** `int smi_ospf_get_area_aggregate_lsdh_type (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int type, struct pal_in4_addr addr, struct pal_in4_addr mask, int * areAggrLsdhType, u_int32_t vrId)`

Gets type of the address aggregate. This field specifies the Lsdh type that this address aggregate applies to. smi\_ospf\_get\_area\_aggregate\_lsdh\_type

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>.  
 ← *area\_id* Aggregate Area ID  
 ← *type* Address aggregate type.

- ← *addr* IP address of the Net or Subnet
- ← *mask* Subnet mask
- *areAggrLsdbType* Address aggregate type
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.88** `int smi_ospf_get_area_aggregate_mask (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int type, struct pal_in4_addr addr, struct pal_in4_addr mask, struct pal_in4_addr * areaAggrMask, u_int32_t vrId)`

Gets subnet mask that pertains to the net or subnet. smi\_ospf\_get\_area\_aggregate\_mask

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Aggregate Area ID
- ← *type* Address aggregate type.
- ← *addr* IP address of the Net or Subnet
- ← *mask* Subnet mask
- *areaAggrMask* Subnet mask
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.89** `int smi_ospf_get_area_aggregate_net (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int type, struct pal_in4_addr addr, struct pal_in4_addr mask, struct pal_in4_addr * areaAggrNet, u_int32_t vrId)`

Gets the IP address of the net or subnet indicated by the range. smi\_ospf\_get\_area\_aggregate\_net

**Parameters:**

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

← *ospfProcessId* OSPF process ID <0-65535>.  
 ← *area\_id* Aggregate Area ID  
 ← *type* Address aggregate type.  
 ← *addr* IP address of the Net or Subnet  
 ← *mask* Subnet mask  
 → *areaAggrNet* IP address of the Net or Subnet  
 ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.90** `int smi_ospf_get_area_aggregate_route_tag (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int type, struct pal_in4_addr addr, struct pal_in4_addr mask, int * areaAggregateRouteTag, u_int32_t vrId)`

Gets the external route tag to be included in NSSA (type-7). smi\_ospf\_get\_area\_aggregate\_route\_tag

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>.  
 ← *area\_id* Aggregate Area ID  
 ← *type* Address aggregate type.  
 ← *addr* IP address of the Net or Subnet  
 ← *mask* Subnet mask  
 → *area\_aggregate\_route\_tag* external route tag to be included in NSSA (type-7)  
 ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.91** `int smi_ospf_get_area_aggregate_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int type, struct pal_in4_addr addr, struct pal_in4_addr mask, int * areaAggrStatus, u_int32_t vrId)`

Gets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction. smi\_ospf\_get\_area\_aggregate\_status

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Aggregate Area ID
- ← *type* Address aggregate type.
- ← *addr* IP address of the Net or Subnet
- ← *mask* Subnet mask
- *areaAggrStatus* Table management action status
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.92 int smi\_ospf\_get\_area\_bdr\_rtr\_count (struct smiclient\_globals \* azg, int ospfProcessId, struct pal\_in4\_addr areaId, int \* areaBdrRouterCount, u\_int32\_t vrId)**

Gets the 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\_ospf\_get\_area\_bdr\_rtr\_count.

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- *areaBdrRouterCount* Counter
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.93 int smi\_ospf\_get\_area\_bdr\_rtr\_status (struct smiclient\_globals \* azg, int ospfProcessId, int \* areaBdrRouterStatus, u\_int32\_t vrId)**

Gets a flag to note whether this router is an Area Border Router. smi\_ospf\_get\_area\_bdr\_rtr\_status

**Parameters:**

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



← *ospfProcessId* OSPF process ID <0-65535>

→ *status* Enabled/disabled

← *vrId* Virtual Router ID

#### Returns:

0 on success, otherwise one of the following error codes

**2.1.2.94** `int smi_ospf_get_area_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr * retAreaId, u_int32_t vrId)`

Gets the 32-bit integer uniquely identifying an area. Area ID 0.0.0.0 is used for the OSPF backbone. `smi_ospf_get_area_id`

#### Parameters:

← *azg* Pointer to the SMI client global structure

← *ospfProcessId* OSPF process ID <0-65535>

← *areaId* Area ID

→ *retAreaId* Area ID

← *vrId* Virtual Router ID

#### Returns:

0 on success, otherwise one of the following error codes

OSPF\_API\_GET\_ERROR

**2.1.2.95** `int smi_ospf_get_area_lsa_cksum_sum (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int * areaLsaCksumSum, u_int32_t vrId)`

Gets 32-bit sum of the link state advertisements' LS checksums contained in this area's link state database. This sum excludes external (LS type-5) link state advertisements. The sum can be used to determine if there has been a change in a router's link state database, and to compare the link state database of two routers. The value should be treated as unsigned when comparing two sums of checksums. `smi_ospf_get_area_lsa_cksum_sum`

#### Parameters:

← *azg* Pointer to the SMI client global structure

← *ospfProcessId* OSPF process ID <0-65535>

← *areaId* Area ID

→ *areaLsaCksumSum* Sum of LSAs's LS checksums

← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.96** `int smi_ospf_get_area_lsa_count (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int * areaLsaCount, u_int32_t vrId)`

Gets the total number of link state advertisements in this area's link state database, excluding AS-external LSAs. `smi_ospf_get_area_lsa_count`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* Area ID  
 → *areaLsaCount* Counter  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.97** `int smi_ospf_get_area_lsa_count_number (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int * areaLsaCountNumber, u_int32_t vrId)`

Gets the total number of link state advertisements in this area's link state database, excluding AS-external LSAs. `smi_ospf_get_area_lsa_count_number`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *area\_id* Area ID  
 → *area\_lsa\_count\_number* Counter  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.98** `int smi_ospf_get_area_nssa_translator_events (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int * nssaTranlatorChangeCnt, u_int32_t vrid)`

Gets an object that indicates the number of translator state changes that have occurred since the last boot-up. smi\_ospf\_get\_area\_nssa\_translator\_events

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- *nssaTranlatorChangeCnt* Counter
- ← *vrid* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.99** `int smi_ospf_get_area_nssa_translator_role (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int * nssaTranlatorRole, u_int32_t vrid)`

Gets an object that indicates an NSSA border router's ability to perform NSSA translation of type-7 LSAs into type-5 LSAs. smi\_ospf\_get\_area\_nssa\_translator\_role

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- *nssaTranRole* Translator role (always/candidate)
- ← *vrid* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.100** `int smi_ospf_get_area_nssa_translator_stability_interval (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int * nssaStabilityInterval, u_int32_t vrid)`

Gets 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\_ospf\_get\_area\_nssa\_translator\_stability\_interval

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- *nssaStabilityInterval* NSSA stability interval in seconds
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.101** `int smi_ospf_get_area_nssa_translator_state (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int * nssaTranslatorState, u_int32_t vrId)`

Gets an object that indicates if and how an NSSA border router is performing NSSA translation of type-7 LSAs into type-5 LSAs. When this object is set to enabled, the NSSA Border router's OspfAreaNssaExtTranslatorRole has been set to always. When this object is set to elected, a candidate NSSA Border router is Translating type-7 LSAs into type-5. When this object is set to disabled, a candidate NSSA border router is NOT translating type-7 LSAs into type-5. smi\_ospf\_get\_area\_nssa\_translator\_state

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- *nssaTranslatorState* Translator state (enabled/elected/disabled)
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.102** `int smi_ospf_get_area_range_area_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr ospfAreaRange, struct pal_in4_addr * areaRangeAreaId, u_int32_t vrId)`

Gets the area that the address range is to be found within. smi\_ospf\_get\_area\_range\_area\_id

**Parameters:**

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

← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* Area ID  
 ← *ospfAreaRange* IP address of the net or subnet  
 → *areaRangeAreaId* Area ID  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.103** `int smi_ospf_get_area_range_effect (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr ospfAreaRange, int * areaRangeEffect, u_int32_t vrId)`

Gets the object that permits management of subnets subsumed by ranges either trigger the advertisement of the indicated summary (advertiseMatching) or result in the subnet's not being advertised at all outside the area. smi\_ospf\_get\_area\_range\_effect

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *area\_id* Area ID  
 ← *ospfAreaRange* IP address of the net or subnet  
 → *areaRangeEffect* Management object for subnets  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_INCONSISTENT\_VALUE  
 OSPF\_API\_SET\_ERR\_INCONSISTENT\_VALUE  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE  
 OSPF\_API\_GET\_ERROR

**2.1.2.104** `int smi_ospf_get_area_range_mask (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr ospfAreaRange, struct pal_in4_addr * areaRangeMask, u_int32_t vrId)`

Gets the subnet mask that pertains to the net or subnet. smi\_ospf\_get\_area\_range\_mask

**Parameters:**

← *azg* Pointer to the SMI client global structure

← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* Area ID  
 ← *ospfAreaRange* IP address of the net or subnet  
 → *areaRangeMask* Subnet mask  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.105** `int smi_ospf_get_area_range_net (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr ospfAreaRange, struct pal_in4_addr * areaRangeNet, u_int32_t vrId)`

Gets the IP address of the net or subnet indicated by the range. smi\_ospf\_get\_area\_range\_net

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* Area ID  
 ← *ospfAreaRange* IP address of the net or subnet  
 → *areaRangeNet* IP address of the net or subnet  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.106** `int smi_ospf_get_area_range_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr ospfAreaRange, int * areaRangeStatus, u_int32_t vrId)`

Gets the object that permits management of the table by facilitating actions such as row creation, construction, and destruction. smi\_ospf\_get\_area\_range\_status

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *area\_id* Area ID  
 ← *ospfAreaRange* IP address of the net or subnet

→ *areaRangeStatus* Table management action status (create/construct/destroy)  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.107** `int smi_ospf_get_area_status (struct smiclient_globals * azg, int  
 ospfProcessId, struct pal_in4_addr areaId, int * areaStatus,  
 u_int32_t vrId)`

Gets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction. smi\_ospf\_get\_area\_status

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* Area ID  
 → *areaStatus* Table management action status (create/construct/destroy)  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.108** `int smi_ospf_get_area_summary (struct smiclient_globals * azg,  
 int ospfProcessId, struct pal_in4_addr areaId, int * areaSummary,  
 u_int32_t vrId)`

Gets a flag that controls the import of summary LSAs into stub and NSSA areas. It has no effect on other areas. If it is noAreaSummary, the router will not originate summary LSAs into the stub or NSSA area. It will rely entirely on its default route. If it is sendAreaSummary, the router will both summarize and propagate summary LSAs. smi\_ospf\_get\_area\_summary

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* Area ID  
 → *areaSummary* Flag (noAreaSummary/sendAreaSummary)  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.109** `int smi_ospf_get_as_lsdb_age (struct smiclient_globals * azg, int ospfProcessId, int lsaType, struct pal_in4_addr lsid, struct pal_in4_addr routerId, int * as_lsdb_age, u_int32_t vrId)`

Gets the age of the LSA in seconds. `smi_ospf_get_as_lsdb_age`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *as* type LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *as\_lsdb\_age* LSA age.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.110** `int smi_ospf_get_as_lsdb_checksum (struct smiclient_globals * azg, int ospfProcessId, int lsaType, struct pal_in4_addr lsid, struct pal_in4_addr routerId, int * as_lsdb_checksum, u_int32_t vrId)`

Gets the 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. The checksum used is the same that is used for ISO connectionless datagrams; it is commonly referred to as the Fletcher checksum. `smi_ospf_get_as_lsdb_checksum`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* AS LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *as\_lsdb\_checksum* LSA checksum.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR



**2.1.2.111** `int smi_ospf_get_as_lsdb_sequence (struct smiclient_globals * azg,  
int ospfProcessId, int lsaType, struct pal_in4_addr lsid, struct  
pal_in4_addr routerId, int * as_lsdb_sequence, u_int32_t vrId)`

Gets the sequence number field is a signed 32-bit integer. It starts with the value '80000001'h, or -'7FFFFFFF'h, and increments until '7FFFFFFF'h. Thus, a typical sequence number will be very negative. It is used to detect old and duplicate link state advertisements. The space of sequence numbers is linearly ordered. The larger the sequence number, the more recent the advertisement. `smi_ospf_get_as_lsdb_sequence`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* AS LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *as\_lsdb\_sequence* Sequence number.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.112** `int smi_ospf_get_as_scope_lsa_count (struct smiclient_globals * azg,  
int ospfProcessId, int * asScopeLsaCount, u_int32_t vrId)`

Gets the number of AS-scope link state in the AS-scope link state database. `smi_ospf_get_as_scope_lsa_count`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *as\_scope\_lsa\_count* Count of AS-scope LSA
- ← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.113** `int smi_ospf_get_asbdr_rtr_count (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int * areaASBDRRouterCount, u_int32_t vrId)`

Gets the total number of Autonomous System Border Routers reachable within this area. This is initially zero and is calculated in each Shortest Path First (SPF) pass. `smi_ospf_get_asbdr_rtr_count`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- *areaASBDRRouterCount* Counter
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.114** `int smi_ospf_get_asbdr_rtr_status (struct smiclient_globals * azg, int ospfProcessId, int * areaASBDRRouterStatus, u_int32_t vrId)`

Gets a flag to note whether this router is configured as an Autonomous System Border Router. `smi_ospf_get_asbdr_rtr_status`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *areaASBDRRouterStatus* Enabled/disabled (1|2)
- ← *vrId* Virtual Router ID

**Returns:**

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

**2.1.2.115** `int smi_ospf_get_auth_type (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int * areaAuthType, u_int32_t vrId)`

Gets the authentication type specified for an area. `smi_ospf_get_auth_type`

**Parameters:**

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

← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* Area ID  
 → *areaAuthType* Authentication type (Simple | Cryptographic)  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.116** `int smi_ospf_get_compatible_rfc1583 (struct smiclient_globals * azg, int ospfProcessId, int * compatibleRfc1583, u_int32_t vrId)`

Gets the whether RFC1583Compatibility is enabled or not. default interface metrics. smi\_ospf\_get\_compatible\_rfc1583

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 → *compatible\_rfc1583* Truth Value 1/2  
 ← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.117** `int smi_ospf_get_demand_extensions (struct smiclient_globals * azg, int ospfProcessId, int * demandExtStatus, u_int32_t vrId)`

Gets the flag to note whether this router supports demand routing. smi\_ospf\_get\_demand\_extensions

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 → *demandExtStatus*  
 ← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes

### 2.1.2.118 `int smi_ospf_get_discontinuity_time (struct smiclient_globals * azg, int ospfProcessId, int * retDiscontinuityTime, u_int32_t vrId)`

Gets the value of sysUpTime on the most recent occasion which any one of this MIB.s counters suffered a discontinuity. `smi_ospf_get_discontinuity_time`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *ret\_discontinuity\_time* TimeStamp in seconds
- ← *vrId* Virtual Router ID

#### Note:

Currently this is not supported

#### Returns:

0 on success, otherwise one of the following error codes

### 2.1.2.119 `int smi_ospf_get_domain_id (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct smi_ospf_vrf_domain_id * ret_pdomain_id, struct list * sDomain_list)`

Get the configured domain\_id information. `smi_ospf_get_domain_id`

#### Parameters:

- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- *domain\_id* Domain id info of type struct `smi_ospf_vrf_domain_id`

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_INVALID\_HEX\_VALUE

### 2.1.2.120 `int smi_ospf_get_exit_overflow_interval (struct smiclient_globals * azg, int ospfProcessId, int * exitOvrflwInterval, u_int32_t vrId)`

Gets the number of seconds that, after entering OverflowState, a router will attempt to leave OverflowState. This allows the router to again originate non-default AS-external LSAs.

When set to 0, the router will not leave overflow state until restarted. `smi_ospf_get_exit_overflow_interval`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *exitOvrflwInterval*
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.121** `int smi_ospf_get_ext_lsdb_advertisement (struct smiclient_globals * azg, int ospfProcessId, int type, struct pal_in4_addr lsid, struct pal_in4_addr routerId, struct smi_lsa_header * lsa, size_t * extLsdbAdv, u_int32_t vrId)`

Gets the entire link state advertisement, including its header. `smi_ospf_get_ext_lsdb_advertisement`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* External LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *extLsdbAdv* LSA.
- *size* LSA size.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.122** `int smi_ospf_get_ext_lsdb_age (struct smiclient_globals * azg, int ospfProcessId, int type, struct pal_in4_addr lsid, struct pal_in4_addr routerId, int * extLsdbAge, u_int32_t vrId)`

Gets the age of the LSA in seconds. `smi_ospf_get_ext_lsdb_age`

**Parameters:**

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

← *ospfProcessId* OSPF process ID <0-65535>.  
 ← *type* External LSA type  
 ← *lsid* Link State ID  
 ← *router\_id* Router ID  
 → *extLsdbAge* LSA age.  
 ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

### 2.1.2.123 int smi\_ospf\_get\_ext\_lsdb\_checksum (struct smiclient\_globals \* azg, int ospfProcessId, int type, struct pal\_in4\_addr lsid, struct pal\_in4\_addr routerId, int \* extLsdbChecksum, u\_int32\_t vrId)

Gets the 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. The checksum used is the same that is used for ISO connectionless datagrams; it is commonly referred to as the Fletcher checksum. smi\_ospf\_get\_ext\_lsdb\_checksum

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>.  
 ← *type* External LSA type  
 ← *lsid* Link State ID  
 ← *router\_id* Router ID  
 → *extLsdbChecksum* LSA checksum.  
 ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

### 2.1.2.124 int smi\_ospf\_get\_ext\_lsdb\_limit (struct smiclient\_globals \* azg, int ospfProcessId, int \* lsdbLimitCount, u\_int32\_t vrId)

Gets the maximum number of non-default AS-external LSAs entries that can be stored in the link state database.

If the value is -1, then there is no limit.

When the number of non-default AS-external LSAs in a router's link state database reaches `ospfExtLsdbLimit`, the router enters overflow state.

The router never holds more than `ospfExtLsdbLimit` non-default AS-external LSAs in its database.

`ospfExtLsdbLimit` MUST be set identically in all routers attached to the OSPF backbone and/or any regular OSPF area (i.e., OSPF stub areas and NSSAs are excluded).  
`smi_ospf_get_ext_lsdb_limit`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *lsdbLimitCount* Counter
- ← *vrId* Virtual Router ID

#### Returns:

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

**2.1.2.125** `int smi_ospf_get_ext_lsdb_lsid (struct smiclient_globals * azg, int ospfProcessId, int type, struct pal_in4_addr lsid, struct pal_in4_addr routerId, struct pal_in4_addr * extLsdbLsid, u_int32_t vrId)`

Gets the link state ID. This ID is an LS Type Specific field containing either a Router ID or an IP Address; it identifies the piece of the routing domain that is being described by the advertisement. `smi_ospf_get_ext_lsdb_lsid`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* External LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID for Advertising router
- *extLsdbLsid* Link State ID
- ← *vrId* Virtual Router Id

#### Returns:

- OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.126** `int smi_ospf_get_ext_lsdb_router_id (struct smiclient_globals * azg, int ospfProcessId, int type, struct pal_in4_addr lsid, struct pal_in4_addr routerId, struct pal_in4_addr * extLsdbRouterId, u_int32_t vrId)`

Gets the 32-bit number that uniquely identifies the originating router in the Autonomous System. `smi_ospf_get_ext_lsdb_router_id`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* External LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *extLsdbRouterId* ID of the advertised router.
- ← *vrId* Virtual Router Id

**Returns:**

- OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.127** `int smi_ospf_get_ext_lsdb_sequence (struct smiclient_globals * azg, int ospfProcessId, int type, struct pal_in4_addr lsid, struct pal_in4_addr routerId, int * extLsdbSequence, u_int32_t vrId)`

Gets the sequence number field is a signed 32-bit integer. It starts with the value '80000001'h, or '-7FFFFFFF'h, and increments until '7FFFFFFF'h. Thus, a typical sequence number will be very negative. It is used to detect old and duplicate link state advertisements. The space of sequence numbers is linearly ordered. The larger the sequence number, the more recent the advertisement. `smi_ospf_get_ext_lsdb_sequence`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* External LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *extLsdbSequence* Sequence number.
- ← *vrId* Virtual Router Id

**Returns:**

- OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR



**2.1.2.128** `int smi_ospf_get_ext_lsdb_type (struct smiclient_globals * azg, int ospfProcessId, int type, struct pal_in4_addr lsid, struct pal_in4_addr routerId, int * extLsdbType, u_int32_t vrId)`

Gets the LSA type. Each type has a distinct advertising format. `smi_ospf_get_ext_lsdb_type`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* External LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID for Advertising router
- *extLsdbType* The LSA type.
- ← *vrId* Virtual Router Id

**Returns:**

- OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.129** `int smi_ospf_get_extern_lsa_cksum_sum (struct smiclient_globals * azg, int ospfProcessId, int * lsaChecksumSum, u_int32_t vrId)`

Gets the 32-bit sum of the LS checksums of the external link state advertisements contained in the link state database.

This sum can be used to determine if there has been a change in a router's link state database and to compare the link state database of two routers.

The value should be treated as unsigned when comparing two sums of checksums. `smi_ospf_get_extern_lsa_cksum_sum`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *lsaChecksumSum* Sum of LS checksums of external LSAs
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

### 2.1.2.130 `int smi_ospf_get_extern_lsa_count (struct smiclient_globals * azg, int ospfProcessId, int * lsaCount, u_int32_t vrId)`

Gets the number of external (LS type-5) link state advertisements in the link state database. `smi_ospf_get_extern_lsa_count`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *lsaCount* Count
- ← *vrId* Virtual Router ID

#### Returns:

0 on success, otherwise one of the following error codes

### 2.1.2.131 `int smi_ospf_get_external_type1_metric (struct smiclient_globals * azg, int ospfProcessId, int * externalType1Metric, u_int32_t vrId)`

Gets the inter area metric value. `smi_ospf_get_external_type1_metric`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *external\_type1\_metric* external type1 metric
- ← *vrId* Virtual Router ID

#### Returns:

0 on success, otherwise one of the following error codes

### 2.1.2.132 `int smi_ospf_get_external_type2_metric (struct smiclient_globals * azg, int ospfProcessId, int * externalType2Metric, u_int32_t vrId)`

Gets the inter area metric value. `smi_ospf_get_external_type2_metric`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *external\_type2\_metric* external type2 metric
- ← *vrId* Virtual Router ID

#### Returns:

0 on success, otherwise one of the following error codes

**2.1.2.133** `int smi_ospf_get_host_area_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr hostAddr, int typeOfService, struct pal_in4_addr * hostAreaId, u_int32_t vrId)`

Gets the OSPF area to which the host belongs. Deprecated by ospfHostCfgAreaID. facilitating actions such as row creation, construction, and destruction. smi\_ospf\_get\_host\_area\_id

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *hostAddr* Host address
- ← *typeOfService* Type of Service
- *hostAreaId* Table management action status (create/construct/destroy)
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.134** `int smi_ospf_get_host_cfg_area_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr addr, int tos, struct pal_in4_addr * hostCfgAreaId, u_int32_t vrId)`

Gets the OSPF area to which the host belongs. facilitating actions such as row creation, construction, and destruction. smi\_ospf\_get\_host\_cfg\_area\_id

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* Host address
- ← *tos* Type of Service
- *host\_cfg\_area\_id* Table management action status (create/construct/destroy)
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.135** `int smi_ospf_get_host_ip_address (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int typeOfService, struct pal_in4_addr * hostIpAddr, u_int32_t vrId)`

Gets the IP address of the host. `smi_ospf_get_host_ip_address`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *hostAddr* Host address
- ← *typeOfService* Type of Service
- *hostIpAddr* Host address
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.136** `int smi_ospf_get_host_metric (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr hostAddr, int typeOfService, int * hostMetric, u_int32_t vrId)`

Gets the metric to be advertised. `smi_ospf_get_host_metric`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *hostAddr* Host address
- ← *typeOfService* Type of Service
- *hostMetric* Metric to be advertised
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.137** `int smi_ospf_get_host_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr hostAddr, int typeOfService, int * hostStatus, u_int32_t vrId)`

Gets the object that permits management of the table by facilitating actions such as row creation, construction, and destruction. `smi_ospf_get_host_status`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *hostAddr* Host address
- ← *typeOfService* Type of Service
- *hostStatus* Table management action status (create/construct/destruct)
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.138** `int smi_ospf_get_host_tos (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr hostAddr, int typeOfService, int * hostTypeOfService, u_int32_t vrId)`

Gets the Type of Service of the route being configured. smi\_ospf\_get\_host\_tos

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *hostAddr* Host address
- ← *typeOfService* Type of Service
- *hostTyteOfService* Type of Service
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.139** `int smi_ospf_get_if_admin_stat (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int * ifAdminStat, u_int32_t vrId)`

Gets the administrative status of the OSPF interface. The value formed on the interface, and the interface will be advertised as an internal route to some area. The value 'disabled' denotes that the interface is external to OSPF. smi\_ospf\_get\_if\_admin\_stat

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>

- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifAdminStat* OSPF interface administrative status
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.140** `int smi_ospf_get_if_area_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, struct pal_in4_addr * areaId, u_int32_t vrId)`

Gets the 32-bit integer uniquely identifying the area to which the interface connects. Area ID 0.0.0.0 is used for the OSPF backbone. `smi_ospf_get_if_area_id`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *areaId* Area ID
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.141** `int smi_ospf_get_if_auth_key (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, char * ifAuthKey, u_int32_t vrId)`

Gets the OSPF authentication key. `smi_ospf_get_if_auth_key`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero

→ *ifAuthKey* An octet string of length zero

← *vrId* Virtual Router ID

#### Returns:

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_GET\_ERROR

**2.1.2.142** `int smi_ospf_get_if_auth_type (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int * ifAuthType, u_int32_t vrId)`

Gets the authentication type specified for an interface. smi\_ospf\_get\_if\_auth\_type

#### Parameters:

← *azg* Pointer to the SMI client global structure

← *ospfProcessId* OSPF process ID <0-65535>

← *ipAddr* IP address of this OSPF interface

← *ifIndex* Interface index or zero

→ *ifAuthType* OSPF authentication type <0-2> none (0),  
simplePassword (1),  
md5 (2)

← *vrId* Virtual Router ID

#### Returns:

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_GET\_ERROR

**2.1.2.143** `int smi_ospf_get_if_backup_designated_router (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, struct pal_in4_addr * ifBkpDesigRouter, u_int32_t vrId)`

Gets the IP address of the back-up designated router. smi\_ospf\_get\_if\_backup\_designated\_router

#### Parameters:

← *azg* Pointer to the SMI client global structure

← *ospfProcessId* OSPF process ID <0-65535>

← *ipAddr* IP address of this OSPF interface

← *ifIndex* Interface index or zero

→ *ifBkpDesigRouter* IP address of the back-up designated router  
 ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.144** `int smi_ospf_get_if_bdr (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr addr, int ifIndex, struct pal_in4_addr * ifBdr, u_int32_t vrId)`

Gets the get the DR id. smi\_ospf\_get\_if\_bdr

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *addr* IP address of this OSPF interface  
 ← *ifIndex* Interface index or zero  
 → *if\_bdr* the DR id.  
 ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.145** `int smi_ospf_get_if_demand (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int * ifDemand, u_int32_t vrId)`

Gets the variable that indicates whether Demand OSPF procedures (hello suppression to FULL neighbors and setting the DoNotAge flag on propagated LSAs) should be performed on the interface. smi\_ospf\_get\_if\_demand

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *ipAddr* IP address of this OSPF interface  
 ← *ifIndex* Interface index or zero  
 → *ifDemand* Demand OSPF procedures flag (True/False)



← *vrId* Virtual Router ID

#### Returns:

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.146** `int smi_ospf_get_if_designated_router (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, struct pal_in4_addr * ifDesigRouter, u_int32_t vrId)`

Gets the IP address of the designated router. smi\_ospf\_get\_if\_designated\_router

#### Parameters:

← *azg* Pointer to the SMI client global structure  
← *ospfProcessId* OSPF process ID <0-65535>  
← *ipAddr* IP address of this OSPF interface  
← *ifIndex* Interface index or zero  
→ *ifDesigRouter* IP address of the designated router  
← *vrId* Virtual Router ID

#### Returns:

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.147** `int smi_ospf_get_if_dr (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr addr, int ifIndex, struct pal_in4_addr * ifDr, u_int32_t vrId)`

Gets the get the DR id. smi\_ospf\_get\_if\_dr

#### Parameters:

← *azg* Pointer to the SMI client global structure  
← *ospfProcessId* OSPF process ID <0-65535>  
← *addr* IP address of this OSPF interface  
← *ifIndex* Interface index or zero  
→ *if\_dr* the DR id.  
← *vrId* Virtual Router ID

#### Returns:

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.148** `int smi_ospf_get_if_events (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int * ifEvents, u_int32_t vrId)`

Gets the of times the interface has changed state and an error has occurred. `smi_ospf_get_if_events`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifEvents* The number of events
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.149** `int smi_ospf_get_if_hello_interval (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int * ifHelloInterval, u_int32_t vrId)`

Gets the interval, in seconds, between the Hello packets that the router sends on the interface. This value must be the same for all routers attached to a common network. `smi_ospf_get_if_hello_interval`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifHelloInterval* Hello interval
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.150** `int smi_ospf_get_if_ip_address (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, struct pal_in4_addr * ifIpAddr, u_int32_t vrId)`

Gets the IP address of this OSPF interface. `smi_ospf_get_if_ip_address`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifIpAddr* IP address of this OSPF interface
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.151** `int smi_ospf_get_if_lsa_checksum (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr addr, int ifIndex, int * ifLsaChecksum, u_int32_t vrId)`

Gets the get the checksum of the complete contents of the advertisement, excepting the age field. `smi_ospf_get_if_lsa_checksum`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *if\_lsa\_checksum* the checksum of the complete contents of the advertisement, excepting the age field.
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.152** `int smi_ospf_get_if_lsa_count (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr addr, int ifIndex, int *  
if_lsa_count, u_int32_t vrId)`

Gets the LSA count for an interface. `smi_ospf_get_if_lsa_count`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *if\_lsa\_count* the LSA count for an interface.
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.153** `int smi_ospf_get_if_metric_address_less_if (struct smiclient_globals *  
azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int  
tos, int * ifMetricAddrLessif, u_int32_t vrId)`

Gets the addressless IF index for interfaces that do not have an IP address. This facilitates the creation of instances of addressed and address-less interfaces. This variable takes the value 0 on interfaces with IP Addresses, and the value of *ifIndex* for interfaces having no IP Address. On row creation, this value can be derived from the instance. `smi_ospf_get_if_metric_address_less_if`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *addr* IP address of this OSPF interface.
- ← *ifIndex* Interface index or zero.
- ← *tos* Type of Service.
- *ifMetricAddrLessif* The OSPF interface index.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.154** `int smi_ospf_get_if_metric_ip_address (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int typeOfService, struct pal_in4_addr * ifMetricIpAddr, u_int32_t vrId)`

Gets the IP address of this OSPF interface. smi\_ospf\_get\_if\_metric\_ip\_address

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero.
- ← *typeOfService* Type of Service
- *ifMetricIpAddr* OSPF interface address
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.155** `int smi_ospf_get_if_metric_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int typeOfService, int * ifMetricStatus, u_int32_t vrId)`

Gets object permits management of the table by facilitating actions such as row creation, construction, and destruction. smi\_ospf\_get\_if\_metric\_status

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ipAddr* IP address of this OSPF interface.
- ← *ifIndex* Interface index or zero.
- ← *typeOfService* Type of Service.
- *ifMetricStatus* Table management action status.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.156** `int smi_ospf_get_if_metric_value (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int typeOfService, int * ifMetricValue, u_int32_t vrId)`

Gets the metric of using this TOS on this interface. `smi_ospf_get_if_metric_value`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ipAddr* IP address of this OSPF interface.
- ← *ifIndex* Interface index or zero.
- ← *typeOfService* Type of Service.
- *ifMetricValue* Metric value
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.157** `int smi_ospf_get_if_multicast_forwarding (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int * ifMulticastForward, u_int32_t vrId)`

Gets the way multicasts should be forwarded on this interface: not forwarded, forwarded as data link multicasts, or forwarded as data link unicasts. Data link multicasting is not meaningful on point-to-point and nbma interfaces, and setting `ospfMulticastForwarding` to 0 effectively disables all multicast forwarding. `smi_ospf_get_if_multicast_forwarding`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifMulticastForward* Way of multicast forwarding <1-3> blocked (1), -- no multicast forwarding  
multicast (2), -- using multicast address  
unicast (3) -- to each OSPF neighbor
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.158** `int smi_ospf_get_if_poll_interval (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int *  
ifPollInterval, u_int32_t vrId)`

Gets the interval, in seconds, between the Hello packets sent to an inactive non-broadcast multi-access neighbor. `smi_ospf_get_if_poll_interval`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifPollInterval* Poll interval
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.159** `int smi_ospf_get_if_retrans_interval (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int *  
ifRetransmitInterval, u_int32_t vrId)`

Gets the interval, in seconds, between link-state-advertisement retransmissions, for adjacencies belonging to this interface. This value is also used when retransmitting database description and Link State request packets. Note that minimal value SHOULD be 1 second. `smi_ospf_get_if_retrans_interval`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifRetransmitInterval* The retransmission interval
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.160** `int smi_ospf_get_if_rtr_dead_interval (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int *  
ifDeadInterval, u_int32_t vrId)`

Gets the dead interval, in seconds, that a router's Hello packets have not been seen before its neighbors declare the router down. This should be some multiple of the Hello interval. This value must be the same for all routers attached to a common network.  
smi\_ospf\_get\_if\_rtr\_dead\_interval

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifDeadInterval* Dead interval
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.161** `int smi_ospf_get_if_rtr_priority (struct smiclient_globals * azg, int  
ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int *  
ifRouterPriority, u_int32_t vrId)`

Gets the priority of this interface. Used in multi-access networks, this field is used in the designated router election algorithm. The value 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, routers will use their Router ID as a tie breaker. smi\_ospf\_get\_if\_rtr\_priority

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifRouterPriority* Priority of this interface
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR



**2.1.2.162** `int smi_ospf_get_if_state (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int * ifstate, u_int32_t vrId)`

Gets the OSPF interface state. `smi_ospf_get_if_state`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifstate* Interface state <1-7> down (1),  
loopback (2),  
waiting (3),  
pointToPoint (4),  
designatedRouter (5),  
backupDesignatedRouter (6),  
otherDesignatedRouter (7)
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.163** `int smi_ospf_get_if_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int * ifStatus, u_int32_t vrId)`

Gets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction. `smi_ospf_get_if_status`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifStatus* Table management action status
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.164** `int smi_ospf_get_if_transit_delay (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr addr, int ifIndex, int *  
ifTransmitDelay, u_int32_t vrId)`

Gets the transit-delay value of the OSPF interface, which is an estimate of the number of seconds required to transmit a link-state update packet through this interface. Note that the minimal value SHOULD be 1 second. `smi_ospf_get_if_transit_delay`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifTransmitDelay* Transit delay
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.165** `int smi_ospf_get_if_type (struct smiclient_globals * azg, int  
ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int * ifType,  
u_int32_t vrId)`

Gets the OSPF interface type. By way of a default, this field may be intuited from the corresponding value of *ifType*. Broadcast LANs, such as Ethernet and IEEE 802.5, take the value 'broadcast', X.25 and similar technologies take the value 'nbma', and links that are definitively point to point take the value 'pointToPoint'. `smi_ospf_get_if_type`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- *ifType* OSPF interface type
  - broadcast (1),
  - nbma (2),
  - pointToPoint (3),
  - pointToMultipoint (5)
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.166** `int smi_ospf_get_import_as_extern (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr areaId, int * areaType,  
u_int32_t vrId)`

Gets the OSPF area type that indicates if an area is a stub area, NSSA, or standard area. Type-5 AS-external LSAs and type-11 Opaque LSAs are not imported into stub areas or NSSAs. NSSAs import AS-external data as type-7 LSAs. `smi_ospf_get_import_as_extern`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *ospfProcessId* OSPF process ID <0-65535>  
← *areaId* Area ID  
→ *areaType* Area type  
← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.167** `int smi_ospf_get_inter_area_metric (struct smiclient_globals * azg,  
int ospfProcessId, int * interAreaMetric, u_int32_t vrId)`

Gets the inter area metric value. `smi_ospf_get_inter_area_metric`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *ospfProcessId* OSPF process ID <0-65535>  
→ *inter\_area\_metric* inter area metric value  
← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.168** `int smi_ospf_get_intra_area_metric (struct smiclient_globals * azg,  
int ospfProcessId, int * intraAreaMetric, u_int32_t vrId)`

Gets the inter area metric value. `smi_ospf_get_intra_area_metric`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *intra\_area\_metric* intra area metric value
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.169** `int smi_ospf_get_local_lsdB_age (struct smiclient_globals *  
azg, int ospfProcessId, struct pal_in4_addr lsdbIp, signed int  
locLsdBAddLeesIf, int type, struct pal_in4_addr lsid, struct  
pal_in4_addr routerId, int * localLsdBAge, u_int32_t vrId)`

Gets the age of the LSA in seconds. `smi_ospf_get_local_lsdB_age`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* LINK LOCAL LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *local\_lsdB\_age* LSA age.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.170** `int smi_ospf_get_local_lsdB_checksum (struct smiclient_globals *  
azg, int ospfProcessId, struct pal_in4_addr lsdbIp, signed int  
locLsdBAddLeesIf, int type, struct pal_in4_addr lsid, struct  
pal_in4_addr routerId, int * localLsdBChecksum, u_int32_t vrId)`

Gets the 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. The checksum used is the same that is used for ISO connectionless datagrams; it is commonly referred to as the Fletcher checksum. `smi_ospf_get_local_lsdB_checksum`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* LINK LOCAL LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *local\_lsd\_b\_checksum* LSA checksum.
- ← *vrId* Virtual Router Id

**Returns:**

- OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.171** `int smi_ospf_get_local_lsd_b_sequence (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr lsd_bIp, signed int locLsd_bAddLeesIf, int type, struct pal_in4_addr lsid, struct pal_in4_addr routerId, int * localLsd_bSequence, u_int32_t vrId)`

Gets the sequence number field is a signed 32-bit integer.

It starts with the value '80000001'h, or -'7FFFFFFF'h, and increments until '7FFFFFFF'h. Thus, a typical sequence number will be very negative.

It is used to detect old and duplicate link state advertisements.

The space of sequence numbers is linearly ordered.

The larger the sequence number, the more recent the advertisement. `smi_ospf_get_local_lsd_b_sequence`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* LINK-LOCAL LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *local\_lsd\_b\_sequence* Sequence number.
- ← *vrId* Virtual Router Id

**Returns:**

- OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.172** `int smi_ospf_get_lsdb_advertisement (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int lsaType, struct pal_in4_addr lsId, struct pal_in4_addr routerId, struct smi_lsa_header * lsa, size_t * lsdbAdv, u_int32_t vrId)`

Gets the entire link state advertisement, including its header. Note that for variable length LSAs, SNMP agents may not be able to return the largest string size. `smi_ospf_get_lsdb_advertisement`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- ← *lsaType* LSA type
- ← *lsId* Link State ID
- ← *routerId* Originating router ID in the AS
- *lsa* Entire LSA
- *lsdbAdv* The size of LSA
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.173** `int smi_ospf_get_lsdb_age (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int lsaType, struct pal_in4_addr lsId, struct pal_in4_addr routerId, int * lsdbAge, u_int32_t vrId)`

Gets the age of the link state advertisement in seconds. `smi_ospf_get_lsdb_age`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- ← *lsaType* LSA type
- ← *lsId* Link State ID
- ← *routerId* Originating router ID in the AS
- *lsdbAge* Age of LSA in seconds
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.174** `int smi_ospf_get_lsdb_area_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int lsaType, struct pal_in4_addr lsId, struct pal_in4_addr routerId, struct pal_in4_addr * lsdbAreaId, u_int32_t vrId)`

Gets the 32-bit identifier of the area from which the LSA was received. `smi_ospf_get_lsdb_area_id`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- ← *lsaType* LSA type
- ← *lsId* Link State ID
- ← *routerId* Originating router ID in the AS
- *lsdbAreaId* Area ID
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.175** `int smi_ospf_get_lsdb_checksum (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int lsaType, struct pal_in4_addr lsid, struct pal_in4_addr routerId, int * lsdbChksum, u_int32_t vrId)`

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. The checksum used is the same that is used for ISO connectionless datagrams; it is commonly referred to as the Fletcher checksum. `smi_ospf_get_lsdb_checksum`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- ← *lsaType* LSA type
- ← *lsid* Link State ID
- ← *routerId* Originating router ID in the AS
- *lsdbChksum* Checksum
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.176** `int smi_ospf_get_lsdb_lsid (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int lsaType, struct pal_in4_addr lsId, struct pal_in4_addr routerId, struct pal_in4_addr * lsdbLsid, u_int32_t vrId)`

Gets the Link State ID is an LS Type Specific field containing either a Router ID or an IP address; it identifies the piece of the routing domain that is being described by the advertisement. smi\_ospf\_get\_lsdb\_lsid

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* Area ID  
 ← *lsaType* LSA type  
 ← *lsId* Link State ID  
 ← *routerId* Originating router ID in the AS  
 → *lsdbLsid* Link State ID  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.177** `int smi_ospf_get_lsdb_router_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int lsaType, struct pal_in4_addr lsId, struct pal_in4_addr routerId, struct pal_in4_addr * lsdbRouterId, u_int32_t vrId)`

Gets the 32-bit number that uniquely identifies the originating router in the Autonomous System. smi\_ospf\_get\_lsdb\_router\_id

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* Area ID  
 ← *lsaType* LSA type  
 ← *lsId* Link State ID



- ← **routerId** Originating router ID in the AS
- **lsdbRouterId** Originating router ID in the AS
- ← **vrId** Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.178** `int smi_ospf_get_lsdb_sequence (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int lsaType, struct pal_in4_addr lsId, struct pal_in4_addr routerId, int * lsdbSequence, u_int32_t vrId)`

Gets the sequence number field is a signed 32-bit integer. It starts with the value '80000001'h, or -'7FFFFFFF'h, and increments until '7FFFFFFF'h. Thus, a typical sequence number will be very negative. It is used to detect old and duplicate Link State Advertisements. The space of sequence numbers is linearly ordered. The larger the sequence number, the more recent the advertisement. `smi_ospf_get_lsdb_sequence`

**Parameters:**

- ← **azg** Pointer to the SMI client global structure
- ← **ospfProcessId** OSPF process ID <0-65535>
- ← **area\_id** Area ID
- ← **lsaType** LSA type
- ← **lsId** Link State ID
- ← **routerId** Originating router ID in the AS
- **lsdbSequence** Sequence number of LSA
- ← **vrId** Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.179** `int smi_ospf_get_lsdb_type (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int lsaType, struct pal_in4_addr lsId, struct pal_in4_addr routerId, int * lsdbType, u_int32_t vrId)`

Gets the type of the link state advertisement. Each link state type has a separate advertisement format. Note: External link state advertisements are permitted for backward compatibility, but should be displayed in the `ospfAsLsdbTable` rather than here. `smi_ospf_get_lsdb_type`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- ← *lsaType* LSA type
- ← *lsId* Link State ID
- ← *routerId* Originating router ID in the AS
- *lsdbType* LSA type (routerLink| networkLink| summaryLink|  
asSummaryLink| asExternalLink| multicastLink| nssaExternalLink|  
areaOpaqueLink)
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.180** `int smi_ospf_get_nbma_nbr_permanence (struct smiclient_globals *  
azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int  
ifIndex, int * nbmaNeighborPermanence, u_int32_t vrId)`

Gets the variable that displays the status of the entry; 'dynamic' and 'permanent' refer to how the neighbor became known. `smi_ospf_get_nbma_nbr_permanence`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_id* The neighbor ID.
- *nbmaNeighborPermanence* Status of the entry  
dynamic (1), -- learned through protocol  
permanent (2) -- configured address
- ← *vrId* Virtual Router Id

**Returns:**

- OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.181** `int smi_ospf_get_nbma_nbr_status (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex,  
int * nbmaNeighborStatus, u_int32_t vrId)`

Gets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction. `smi_ospf_get_nbma_nbr_status`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_id* The neighbor ID.
- *nbmaNeighborStatus* Table management action status
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.182** `int smi_ospf_get_nbr_address_less_index (struct smiclient_globals *  
azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int  
ifIndex, int * neighborIpAddrLessIndex, u_int32_t vrId)`

Gets the index, on an interface having an IP address, zero. On addressless interfaces, the corresponding value of *ifIndex* in the Internet Standard MIB. On row creation, this can be derived from the instance. `smi_ospf_get_nbr_address_less_index`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_addr* Neighbor IP address.
- *neighborIpAddrLessIndex* Interface index or zero.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.183** `int smi_ospf_get_nbr_events (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, int * nbrEvents, u_int32_t vrId)`

Gets the number of times this neighbor relationship has changed state, or an error has occurred. `smi_ospf_get_nbr_events`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_addr* Neighbor IP address.
- *nbrEvents* Event count
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.184** `int smi_ospf_get_nbr_hello_suppressed (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, int * nbrHelloSuppressed, u_int32_t vrId)`

Gets an indication whether Hellos are being suppressed to the neighbor. `smi_ospf_get_nbr_hello_suppressed`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_id* The neighbor ID.
- *nbrHelloSuppressed* Indication whether Hellos are being suppressed to neighbor
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.185** `int smi_ospf_get_nbr_ip_addr (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, struct pal_in4_addr * nbr_ip_addr, u_int32_t vrId)`

Gets the IP address this neighbor is using in its IP source address. Note that, on addressless links, this will not be 0.0.0.0 but the address of another of the neighbor's interfaces. `smi_ospf_get_nbr_ip_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *nbr\_ip\_addr* Neighbor IP address
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.186** `int smi_ospf_get_nbr_ls_retrans_qlen (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, int * nbrLsRetransQlen, u_int32_t vrId)`

Gets the current length of the retransmission queue. `smi_ospf_get_nbr_ls_retrans_qlen`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *nbrLsRetransQlen* Length of retransmission queue
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.187** `int smi_ospf_get_nbr_options (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, int * nbrOptions, u_int32_t vrId)`

Gets a bit mask corresponding to the neighbor's options field. Bit 0, if set, indicates the system operates on Type of Service metrics other than TOS 0. If zero, the neighbor ignores all metrics, except the TOS 0 metric. Bit 1, if set, indicates the associated area accepts and operates on external information; if zero, it is a stub area. Bit 2, if set, indicates the system routes IP Multicast datagrams, implementing the Multicast Extensions to OSPF. Bit 3, if set, indicates the associated area is an NSSA. These areas carry type 7 external advertisements, which they translate into type 5 external advertisements at NSSA borders. `smi_ospf_get_nbr_options`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_addr* Neighbor IP address.
- *nbrOptions* Neighbor options
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.188** `int smi_ospf_get_nbr_priority (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, int * neighborPriority, u_int32_t vrId)`

Gets the priority of this neighbor in the designated router election algorithm. The value 0 signifies that the neighbor is not eligible to become the designated router on this particular network. `smi_ospf_get_nbr_priority`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_addr* Neighbor IP address.
- *neighborPriority* Neighbor priority
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.189** `int smi_ospf_get_nbr_restart_helper_age (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, int * restartHelperage, u_int32_t vrid)`

Gets the remaining time in current OSPF graceful restartinterval. smi\_ospf\_get\_nbr\_restart\_helper\_age

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *nbr\_addr* Neighbor address.
- ← *ifIndex* Interface index or zero.
- *restart\_helper\_age* indication of remaining time in current OSPF graceful restartinterval
- ← *vrid* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.190** `int smi_ospf_get_nbr_restart_helper_exit_reason (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, int * nbrRestartHelperExitReason, u_int32_t vrid)`

signifies that there has been a change in the graceful restart helper state for the neighbour. This trap should be generated when the neighbour restart helper status transitions for a neighbour smi\_ospf\_get\_nbr\_restart\_helper\_exit\_reason

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *nbr\_addr* Neighbor address.
- ← *ifIndex* Interface index or zero.
- *nbr\_restart\_helper\_exit\_reason* indication that there has been a change in the graceful restart helper state for the neighbour
- ← *vrid* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.191** `int smi_ospf_get_nbr_restart_helper_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, int * nbrRestartHelperStatus, u_int32_t vrid)`

Gets indication whether the router is acting as a graceful restart helper for the neighbor.  
smi\_ospf\_get\_nbr\_restart\_helper\_status

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *nbr\_addr* Neighbor address.
- ← *ifIndex* Interface index or zero.
- *nbr\_restart\_helper\_status* indication whether the router is acting as a graceful restart helper for the neighbor.
- ← *vrid* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.192** `int smi_ospf_get_nbr_rtr_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, struct pal_in4_addr * nbrRouterId, u_int32_t vrid)`

Gets the 32-bit integer (represented as a type IPAddress) uniquely identifying the neighboring router in the Autonomous System. smi\_ospf\_get\_nbr\_rtr\_id

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_addr* Neighbor IP address.
- *nbrRouterId* Neighbor Router ID
- ← *vrid* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR



**2.1.2.193** `int smi_ospf_get_nbr_state (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, int * nbrState, u_int32_t vrId)`

Gets the state of the relationship with the neighbor. `smi_ospf_get_nbr_state`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ifIndex* Interface index or zero.
- ← *nbr\_addr* Neighbor IP address.
- *nbrState* State of relationship with the neighbor <1-8> down (1), attempt (2), init (3), twoway (4), exchangestart (5), exchange (6), loading (7), full (8)
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.194** `int smi_ospf_get_opaque_lsa_support (struct smiclient_globals * azg, int ospfProcessId, int * opaque_lsaSupport, u_int32_t vrId)`

Gets the routers support for OSPF graceful restart. `smi_ospf_get_opaque_lsa_support`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *opaque\_lsa\_support* Truth Value 1/2
- ← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes

### 2.1.2.195 `int smi_ospf_get_originate_new_lsas (struct smiclient_globals * azg, int ospfProcessId, int * lsaCount, u_int32_t vrId)`

Gets the number of new link state advertisements that have been originated from this router.

This number is incremented each time the router originates a new LSA. `smi_ospf_get_originate_new_lsas`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *count* Counter
- ← *vrId* Virtual Router ID

#### Returns:

0 on success, otherwise one of the following error codes

### 2.1.2.196 `int smi_ospf_get_reference_bandwidth (struct smiclient_globals * azg, int ospfProcessId, int * referenceBandwidth, u_int32_t vrId)`

Gets the reference\_bandwidth in Kilobits/sec for default interface metrics. `smi_ospf_get_reference_bandwidth`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *reference\_bandwidth* in Kilobits/sec
- ← *vrId* Virtual Router ID

#### Note:

Currently this is not supported

#### Returns:

0 on success, otherwise one of the following error codes

### 2.1.2.197 `int smi_ospf_get_restart_age (struct smiclient_globals * azg, int ospfProcessId, int * restartAge, u_int32_t vrId)`

Gets the remaining time in current OSPF graceful restart @ interval. `smi_ospf_get_restart_age`

#### Parameters:

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

← *ospfProcessId* OSPF process ID <0-65535>  
→ *restart\_age* restart interval  
← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.198** `int smi_ospf_get_restart_exit_reason (struct smiclient_globals * azg, int ospfProcessId, int * restartExitReason, u_int32_t vrId)`

Gets the outcome of the last attempt at a restart. smi\_ospf\_get\_restart\_exit\_reason

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *ospfProcessId* OSPF process ID <0-65535>  
→ *restart\_exit\_reason* Count of AS-scope LSA  
← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.199** `int smi_ospf_get_restart_interval (struct smiclient_globals * azg, int ospfProcessId, int * time, u_int32_t vrId)`

Gets the OSPF graceful restart timeout interval. smi\_ospf\_get\_restart\_interval

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *ospfProcessId* OSPF process ID <0-65535>  
→ *time* in seconds  
← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes

### 2.1.2.200 `int smi_ospf_get_restart_status (struct smiclient_globals * azg, int ospfProcessId, int * restartStatus, u_int32_t vrId)`

Gets the Current status of OSPF graceful restart. `smi_ospf_get_restart_status`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *restart\_status* (1), (2), (3)
- ← *vrId* Virtual Router ID

#### Note:

Currently this is not supported

#### Returns:

0 on success, otherwise one of the following error codes

### 2.1.2.201 `int smi_ospf_get_restart_strict_lsa_check (struct smiclient_globals * azg, int ospfProcessId, int * restartStrictLsaCheck, u_int32_t vrId)`

Gets the strict LSA checking is enabled for restart. `smi_ospf_get_restart_strict_lsa_check`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *restart\_strict\_lsa\_check* TRUTH-VALUE 1/2
- ← *vrId* Virtual Router ID

#### Note:

Currently this is not supported

#### Returns:

0 on success, otherwise one of the following error codes

### 2.1.2.202 `int smi_ospf_get_restart_support (struct smiclient_globals * azg, int ospfProcessId, int * restartSupport, u_int32_t vrId)`

Gets the routers support for OSPF graceful restart. `smi_ospf_get_restart_support`

#### Parameters:

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

← *ospfProcessId* OSPF process ID <0-65535>

→ *restart\_support* Truth Value 1/2

← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.203 int smi\_ospf\_get\_router\_id (struct smiclient\_globals \* azg, int ospfProcessId, struct pal\_in4\_addr \* routerId, u\_int32\_t vrId)**

Gets the 32-bit integer ID that uniquely identifies the router in the Autonomous System.  
smi\_ospf\_get\_router\_id

**Parameters:**

← *azg* Pointer to the SMI client global structure

← *ospfProcessId* OSPF process ID <0-65535>

→ *routerId* Router ID

← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.204 int smi\_ospf\_get\_rx\_new\_lsas (struct smiclient\_globals \* azg, int ospfProcessId, int \* lsaCount, u\_int32\_t vrId)**

Gets the number of link state advertisements received that are determined to be new instantiations.

This number does not include newer instantiations of self-originated link state advertisements. smi\_ospf\_get\_rx\_new\_lsas

**Parameters:**

← *azg* Pointer to the SMI client global structure

← *ospfProcessId* OSPF process ID <0-65535>

→ *lsaCount* Counter

← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.205** `int smi_ospf_get_settrap (struct smiclient_globals * azg, int ospfProcessId, int * trapFlagBit, u_int32_t vrId)`

Gets the bitmap of traps in ospf. every bit indicates a trap in ospf. smi\_ospf\_get\_settrap

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *trapFlagBit* bitmap of traps in ospf.
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.206** `int smi_ospf_get_spf_runs (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int * spfCount, u_int32_t vrId)`

Gets the number of times that the intra-area route table has been calculated using this area's link state database. smi\_ospf\_get\_spf\_runs

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- *spfCount* Counter
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.207** `int smi_ospf_get_stub_area_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int typeOfService, struct pal_in4_addr * stubAreaId, u_int32_t vrId)`

Gets 32-bit identifier for the stub area. On creation, this can be derived from the instance. smi\_ospf\_get\_stub\_area\_id

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>

- ← *areaId* Area ID
- ← *typeOfService* Type of Service
- *stubAreaId* Stub Area ID
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.208** `int smi_ospf_get_stub_metric (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int typeOfService, int * stubMetric, u_int32_t vrId)`

Gets the metric value applied at the indicated Type of Service. By default, this equals the least metric at the Type of Service among the interfaces to other areas. smi\_ospf\_get\_stub\_metric

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *area\_id* Area ID
- ← *typeOfService* Type of Service
- *stubMetric* Metric cost
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.209** `int smi_ospf_get_stub_metric_type (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int tos, int * stubMetricType, u_int32_t vrId)`

Gets the object that displays the type of metric advertised as a default route. smi\_ospf\_get\_stub\_metric\_type

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- ← *typeOfService* Type of Service

→ *stubMetricType* Metric type (OSPF/External Type1/Type2)  
 ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.210** `int smi_ospf_get_stub_router_advertisement (struct smiclient_globals * azg, int ospfProcessId, int * stubRouterAdvertisement, u_int32_t vrId)`

Gets the number of AS-scope link state in the AS-scope link state database. `smi_ospf_get_stub_router_advertisement`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 → *stub\_router\_advertisement* Count of AS-scope LSA  
 ← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.211** `int smi_ospf_get_stub_router_support (struct smiclient_globals * azg, int ospfProcessId, int * as_scopeLsaCksumsum, u_int32_t vrId)`

Gets the flag to note whether this router supports demand routing. `smi_ospf_get_stub_router_support`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 → *as\_scope\_lsa\_cksumsum* Yes/No  
 ← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes



**2.1.2.212** `int smi_ospf_get_stub_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int typeOfService, int * stubStatus, u_int32_t vrId)`

Gets the object that permits management of the table by facilitating actions such as row creation, construction, and destruction. smi\_ospf\_get\_stub\_status

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- ← *typeOfService* Type of Service
- *stubStatus* Table management action status (create/construct/destroy)
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.213** `int smi_ospf_get_stub_tos (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int typeOfService, int * stubTOS, u_int32_t vrId)`

Gets the Type of Service associated with the metric. On creation, this can be derived from the instance. smi\_ospf\_get\_stub\_tos

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- ← *typeOfService* Type of Service
- *stubTOS* 0
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_GET\_ERROR

**2.1.2.214** `int smi_ospf_get_tos_support (struct smiclient_globals * azg, int ospfProcessId, int * TOSStatus, u_int32_t vrId)`

Gets a flag to note whether router has support for type-of-service routing. smi\_ospf\_get\_tos\_support

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *TOSStatus* (Yes|No)
- ← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.215** `int smi_ospf_get_version_number (struct smiclient_globals * azg, int ospfProcessId, int * ospfVersion, u_int32_t vrId)`

Gets the OSPF version number. `smi_ospf_get_version_number`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- *version* OSPF version (2)
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.216** `int smi_ospf_get_virt_if_area_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, struct pal_in4_addr * virtIfAreaId, u_int32_t vrId)`

Gets the transit area that the virtual link traverses. By definition, this is not 0.0.0.0. `smi_ospf_get_virt_if_area_id`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virtIfAreaId* Transit Area ID.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.217** `int smi_ospf_get_virt_if_auth_key (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, char * virtIfAuthKey, u_int32_t vrId)`

Gets the authentication key. smi\_ospf\_get\_virt\_if\_auth\_key

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>.  
 ← *area\_id* Transit Area ID.  
 ← *peerId* Virtual Neighbor Router ID.  
 → *virtIfAuthKey* Authentication key. An octet string of length zero.  
 ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following i error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.218** `int smi_ospf_get_virt_if_auth_type (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int * virtIfAuthType, u_int32_t vrId)`

Gets the OSPF authentication type of this interface. smi\_ospf\_get\_virt\_if\_auth\_type

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>.  
 ← *area\_id* Transit Area ID.  
 ← *peerId* Virtual Neighbor Router ID.  
 → *virtIfAuthType* Authentication type. none (0),  
     simplePassword (1),  
     md5 (2)  
 ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.219** `int smi_ospf_get_virt_if_events (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int * virtIfEvents, u_int32_t vrId)`

Gets the number of state changes or error events on this virtual link. `smi_ospf_get_virt_if_events`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *peerId* Virtual Neighbor Router ID.
- *virtIfEvents* Event number of state changes or errors
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes

**2.1.2.220** `int smi_ospf_get_virt_if_hello_interval (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int * virtIfHelloIntvl, u_int32_t vrId)`

Gets the hello interval time, in seconds, between the Hello packets the router sends on the interface. This value must be the same for the virtual neighbor. `smi_ospf_get_virt_if_hello_interval`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *peerId* Virtual Neighbor Router ID.
- *virtIfHelloIntvl* Hello interval. The default value is 10.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.221** `int smi_ospf_get_virt_if_lsa_cksumsum (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int * virtLfLsaCksumsum, u_int32_t vrId)`

This function determine if there has been a change in virtual interfaces link state-database,. smi\_ospf\_get\_virt\_if\_lsa\_cksumsum

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virt\_if\_lsa\_cksumsum* 32-bit unsigned sum of the link state . LS checksums contained in this interface.s link-local link state database
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.222** `int smi_ospf_get_virt_if_lsa_count (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr neighborIpAddr, int * virtiflsaCount, u_int32_t vrId)`

This function gets the total number of link-local link state this virtual interfaces link-local link state database. smi\_ospf\_get\_virt\_if\_lsa\_count

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virt\_if\_lsa\_count* number of link-local link state this virtual interfaces.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.223** `int smi_ospf_get_virt_if_neighbor (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, struct pal_in4_addr * virtIfNeighbor, u_int32_t vrId)`

Gets the router ID of the virtual neighbor. smi\_ospf\_get\_virt\_if\_neighbor

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *areaId* Transit Area ID.
- ← *peerId* Virtual Neighbor Router ID.
- *virtIfNeighbor* The router ID of the neighbor.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.224** `int smi_ospf_get_virt_if_retrans_interval (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int * virtIfRetransIntvl, u_int32_t vrId)`

Gets the retransmission interval time, in seconds, between link state advertisement re-transmissions, for adjacencies belonging to this interface. This value is also used when retransmitting database description and Link State request packets. This value should be well over the expected round-trip time. smi\_ospf\_get\_virt\_if\_retrans\_interval

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *areaId* Transit Area ID.
- ← *peerId* Virtual Neighbor Router ID.
- *virtIfRetransIntvl* Retransmission interval. The default value is 5.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.225** `int smi_ospf_get_virt_if_rtr_dead_interval (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int * virtIfRouterDeadIntvl, u_int32_t vrId)`

Gets the dead interval time, in seconds, that a router's Hello packets have not been seen before its neighbors declare the router down. This should be some multiple of the Hello interval. This value must be the same for the virtual neighbor. `smi_ospf_get_virt_if_rtr_dead_interval`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *peerId* Virtual Neighbor Router ID.
- *virtIfRouterDeadIntvl* Dead interval. The default value is 60.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.226** `int smi_ospf_get_virt_if_state (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int * virtIfState, u_int32_t vrId)`

Gets the OSPF virtual interface states. `smi_ospf_get_virt_if_state`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *peerId* Virtual Neighbor Router ID.
- *virtIfState* State.  
down (1), -- these use the same encoding  
pointToPoint (4) -- as the ospfIfTable
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.227** `int smi_ospf_get_virt_if_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int * virtIfStatus, u_int32_t vrId)`

This function gets an object that permits management of the table facilitating actions such as row creation, construction, and destruction. `smi_ospf_get_virt_if_status`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *peerId* Virtual Neighbor Router ID.
- *virtIfStatus* Table management action object.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.228** `int smi_ospf_get_virt_if_transit_delay (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int * virtIfTransitDelay, u_int32_t vrId)`

Gets the estimated number of seconds it takes to transmit a linkstate update packet over this interface. `smi_ospf_get_virt_if_transit_delay`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *areaId* Transit Area ID.
- ← *peerId* Virtual Neighbor Router ID.
- *virtIfTransitDelay* Transit delay.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR



**2.1.2.229** `int smi_ospf_get_virt_local_lsdb_age (struct smiclient_globals  
* azg, int ospfProcessId, struct pal_in4_addr transitArea, struct  
pal_in4_addr neighborIpAddr, int lsaType, struct pal_in4_addr lsid,  
struct pal_in4_addr routerId, int * virt_localLsdbAge, u_int32_t vrId)`

Gets the age of the LSA in seconds. `smi_ospf_get_virt_local_lsdb_age`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* VIRT LINK LOCAL LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *virt\_local\_lsdb\_age* LSA age.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.230** `int smi_ospf_get_virt_local_lsdb_checksum (struct smiclient_globals  
* azg, int proc, struct pal_in4_addr transit_area, struct pal_in4_addr  
neighbor_id, int type, struct pal_in4_addr lsid, struct pal_in4_addr  
routerId, int * virt_local_lsdb_checksum, u_int32_t vrId)`

Gets the 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. The checksum used is the same that is used for ISO connectionless datagrams; it is commonly referred to as the Fletcher checksum. `smi_ospf_get_virt_local_lsdb_checksum`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* VIRT LINK LOCAL LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *virt\_local\_lsdb\_checksum* LSA checksum.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.231** `int smi_ospf_get_virt_local_lsdb_sequence (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr transitArea, struct pal_in4_addr neighborId, int type, struct pal_in4_addr lsid, struct pal_in4_addr routerId, int * virtLocalSsdbSequence, u_int32_t vrId)`

Gets the sequence number field is a signed 32-bit integer. It starts with the value '80000001'h, or -'7FFFFFFF'h, and increments until '7FFFFFFF'h. Thus, a typical sequence number will be very negative. It is used to detect old and duplicate link state advertisements. The space of sequence numbers is linearly ordered. The larger the sequence number, the more recent the advertisement. `smi_ospf_get_virt_local_lsdb_sequence`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *type* VIRT LINK LOCAL LSA type
- ← *lsid* Link State ID
- ← *router\_id* Router ID
- *virt\_local\_lsdb\_sequence* Sequence number.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.232** `int smi_ospf_get_virt_nbr_area (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr neighborIpAddr, struct pal_in4_addr * virtNbrArea, u_int32_t vrId)`

Gets the transit area identifier. `smi_ospf_get_virt_nbr_area`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virtNbrArea* Transit Area ID.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.233** `int smi_ospf_get_virt_nbr_events (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int * virtNbrEvents, u_int32_t vrId)`

Gets the number of times this virtual link has changed its state or an error has occurred.  
smi\_ospf\_get\_virt\_nbr\_events

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virtNbrEvents* Event counter
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.234** `int smi_ospf_get_virt_nbr_hello_suppressed (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int * virtNbrHelloSuppressed, u_int32_t vrId)`

Gets an indication whether Hellos are being suppressed to the neighbor. smi\_ospf\_get\_virt\_nbr\_hello\_suppressed

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virtNbrHelloSuppressed* Hello packet suppresssion state
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.235** `int smi_ospf_get_virt_nbr_ip_addr (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, struct pal_in4_addr * virtNbrIpAddr, u_int32_t vrId)`

Gets the IP address the virtual neighbor is using. `smi_ospf_get_virt_nbr_ip_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virtNbrIpAddr* Virtual Neighbor IP address.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.236** `int smi_ospf_get_virt_nbr_ls_retrans_qlen (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr nbId, int * virtNbrLsRetransQlen, u_int32_t vrId)`

Gets the current length of the retransmission queue. `smi_ospf_get_virt_nbr_ls_retrans_qlen`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virtNbrLsRetransQlen* Retransmission queue length
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.237** `int smi_ospf_get_virt_nbr_options (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int * virtNbrOptions, u_int32_t vrId)`

Gets the A bit mask corresponding to the neighbor's options field. Bit 1, if set, indicates that the system will operate on Type of Service metrics other than ToS 0. If zero, the neighbor will ignore all metrics except the TOS 0 metric. Bit 2, if set, indicates that the system is network multicast capable, i.e., that it implements OSPF multicast routing. smi\_ospf\_get\_virt\_nbr\_options

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *areaId* Transit Area ID.
- ← *nbrId* Virtual Neighbor Router ID.
- *virtNbrOptions* Virtual Neighbor's Options.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.238** `int smi_ospf_get_virt_nbr_restart_helper_age (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int * virtNbrRestartHelperAge, u_int32_t vrId)`

Gets the remaining time in current OSPF graceful restartinterval. smi\_ospf\_get\_virt\_nbr\_restart\_helper\_age

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virt\_nbr\_restart\_helper\_age* indication of remaining time in current OSPF graceful restartinterval
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.239** `int smi_ospf_get_virt_nbr_restart_helper_exit_reason`  
 (struct smiclient\_globals \* *azg*, int *ospfProcessId*, struct  
 pal\_in4\_addr *areaId*, struct pal\_in4\_addr *nbrId*, int \*  
*virtNbrRestartHelperExitReason*, u\_int32\_t *vrId*)

`smi_ospf_get_virt_nbr_restart_helper_exit_reason` brief signifies that there has been a change in the graceful restart helper state for the neighbour. This trap should be generated when the neighbour restart helper status transitions for a neighbour

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virt\_nbr\_restart\_helper\_exit\_reason* indication that there has been a change in the graceful restart helper state for the neighbour
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.240** `int smi_ospf_get_virt_nbr_restart_helper_status` (struct  
 smiclient\_globals \* *azg*, int *ospfProcessId*, struct pal\_in4\_addr  
*areaId*, struct pal\_in4\_addr *nbrId*, int \* *virtNbrRestartHelperStatus*,  
 u\_int32\_t *vrId*)

`smi_ospf_get_virt_nbr_restart_helper_status` brief Gets indication whether the router is acting as a graceful restart helper for the neighbor.

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virt\_nbr\_restart\_helper\_status* indication whether the router is acting as a graceful restart helper for the neighbor.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_GET\_ERROR

**2.1.2.241** `int smi_ospf_get_virt_nbr_rtr_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, struct pal_in4_addr * virtNbrRouterId, u_int32_t vrId)`

Gets the 32-bit integer that uniquely identifies the neighboring router in the Autonomous System. smi\_ospf\_get\_virt\_nbr\_rtr\_id

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virtNbrRouterId* Virtual Neighbor Router ID.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

**2.1.2.242** `int smi_ospf_get_virt_nbr_state (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr nbrId, int * virtNbrState, u_int32_t vrId)`

Gets the state of the virtual neighbor relationship. smi\_ospf\_get\_virt\_nbr\_state

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *area\_id* Transit Area ID.
- ← *nbr\_id* Virtual Neighbor Router ID.
- *virtNbrState* State of the virtual neighbor relationship. down (1), attempt (2), init (3), twoWay (4), exchangeStart (5), exchange (6), loading (7), full (8)
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_GET\_ERROR

### 2.1.2.243 `int smi_ospf_graceful_restart_planned_set_sdkapi (struct smiclient_globals * azg, u_int32_t vrId)`

This function allows to configure only OSPF planned (S/W) restarts. `smi_ospf_graceful_restart_planned_set_sdkapi`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router id

#### Returns:

0 on success

### 2.1.2.244 `int smi_ospf_graceful_restart_planned_unset_sdkapi (struct smiclient_globals * azg, u_int32_t vrId)`

This function allows to configure all types of restarts (Planned and Unplanned). `smi_ospf_graceful_restart_planned_unset_sdkapi`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *om* Pointer to structure `ospf_master`

#### Returns:

0 on success

### 2.1.2.245 `int smi_ospf_graceful_restart_set_sdkapi (struct smiclient_globals * azg, u_int32_t vrId, int restartSeconds)`

Sets the grace period in seconds for OSPF to restart gracefully along with reason for restart. `smi_ospf_graceful_restart_set_sdkapi`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *om* Pointer to structure `ospf_master`
- ← *restartSeconds* Grace period in seconds <1-1800>
- ← *restartReason* Reason for restart (Unknown, Restart, Upgrade and Redundant Switch)

#### Returns:

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_GRACE\_PERIOD\_INVALID  
 OSPF\_API\_SET\_ERR\_INVALID\_REASON  
 OSPF\_API\_SET\_MALLOC\_ERR



**2.1.2.246** `int smi_ospf_graceful_restart_unset_sdkapi (struct smiclient_globals * azg, u_int32_t vrId)`

Disables the OSPF graceful restart method and sets grace period to zero. `smi_ospf_graceful_restart_unset_sdkapi`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *om* Pointer to structure `ospf_master`

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.247** `int smi_ospf_host_entry_cost_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr hostIpAddr, struct pal_in4_addr areaId, int hostEntryCost)`

This function sets stub host entry belonging to particular area along with cost. `smi_ospf_host_entry_cost_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *hostIpAddr* The IP address of the interface.
- ← *areaId* area to which the network belongs.
- ← *hostEntryCost* cost

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_HOST\_ENTRY\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_ID\_NOT\_MATCH  
 OSPF\_API\_SET\_ERR\_HOST\_ENTRY\_NOT\_EXIST

**2.1.2.248** `int smi_ospf_host_entry_cost_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr hostIpAddr, struct pal_in4_addr areaId)`

This function unsets stub host entry belonging to particular area. `smi_ospf_host_entry_cost_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *hostIpAddr* The IP address of the interface.
- ← *area\_id* area to which the network belongs.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_HOST\_ENTRY\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_ID\_NOT\_MATCH

**2.1.2.249** `int smi_ospf_host_entry_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr hostIpAddr, struct pal_in4_addr areaId)`

This sets the ospf stub entry host address. `smi_ospf_host_entry_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *hostIpAddr* The IP address of the interface.
- ← *areaId* area to which the network belongs

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_INVALID\_IPV4\_ADDRESS

**2.1.2.250** `int smi_ospf_host_entry_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr hostIpAddr, struct pal_in4_addr areaId)`

This function removes the ospf stub entry address. `smi_ospf_host_entry_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *hostIpAddr* The IP address of the interface.
- ← *areaId* area to which the network belongs

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_HOST\_ENTRY\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_ID\_NOT\_MATCH

**2.1.2.251** `int smi_ospf_if_authentication_key_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, char * md5, char * ifAuthKey)`

This function sets the authentication key for simple password. `smi_ospf_if_authentication_key_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifname* Name of the interface
- ← *md5* Encryption enabled or not
- ← *ifAuthKey* The authentication key, null-terminated

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.252** `int smi_ospf_if_authentication_key_set_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr, char * md5, char * ifAuthKey)`

This function sets the interface's authentication key for an area of the specified IP address with a simple password. `smi_ospf_if_authentication_key_set_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *ipAddr* The IP address of the interface
- ← *md5* Encryption enabled or not
- ← *ifAuthKey* The authentication key, null-terminated

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.253** `int smi_ospf_if_authentication_key_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

This function removes the interface authentication key for an area. `smi_ospf_if_authentication_key_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.254** `int smi_ospf_if_authentication_key_unset_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr)`

This function disables the authentication key for the interface specified by IP address. `smi_ospf_if_authentication_key_unset_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id

- ← *ifName* Name of the interface
- ← *addr* The IP address of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

### 2.1.2.255 int smi\_ospf\_if\_authentication\_type\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName, u\_char ifAuthType)

Sets This function sets the authentication type of the current . interface. smi\_ospf\_if\_authentication\_type\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name
- ← *ifAuthType* Authentication type ( None | Simple Password | Cryptographic)

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_AUTH\_TYPE\_INVALID  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.256 int smi\_ospf\_if\_authentication\_type\_set\_by\_addr (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName, struct pal\_in4\_addr ipAddr, u\_char ifAuthType)

Sets This function sets the authentication type of the interface specified by IP address. smi\_ospf\_if\_authentication\_type\_set\_by\_addr

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name
- ← *ipAddr* the IP address of the interface.
- ← *ifAuthType* authentication type. ( None | Simple Password | Cryptographic)

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_AUTH\_TYPE\_INVALID  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.257 int smi\_ospf\_if\_authentication\_type\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName)**

Sets This function removes the authentication type for the current interface. smi\_ospf\_if\_authentication\_type\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ifName* The interface name

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.258 int smi\_ospf\_if\_authentication\_type\_unset\_by\_addr (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName, struct pal\_in4\_addr ipAddr)**

Sets This function resets the authentication type for the specified interface. smi\_ospf\_if\_authentication\_type\_unset\_by\_addr

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ifName* The interface name  
 ← *ipAddr* The IP address of the interface.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.259** `int smi_ospf_if_conf_ldp_igp_set_sdkapi (struct smiclient_globals *  
azg, u_int32_t vrId, char * ifName, u_int32_t holdDownTimer)`

Sets LDP-OSPF. smi\_ospf\_if\_conf\_ldp\_igp\_set\_sdkapi

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ifName* Interface Name
- ← *holdDownTimer* HoldDown Time <1-2147483>

**Returns:**

0 on SUCCESS, otherwise one of the following error codes

**2.1.2.260** `int smi_ospf_if_conf_ldp_igp_unset_sdkapi (struct smiclient_globals  
* azg, u_int32_t vrId, char * ifName)`

Unsets LDP-OSPF. smi\_ospf\_if\_conf\_ldp\_igp\_unset\_sdkapi

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *name* Interface Name

**Returns:**

OSPF\_API\_SET\_SUCCESS on SUCCESS, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_IF\_SYNC\_NOT\_EXIST

**2.1.2.261** `int smi_ospf_if_cost_set (struct smiclient_globals * azg, u_int32_t  
vrId, char * ifName, u_int32_t ifCost)`

Sets This function sets the current interface output cost. smi\_ospf\_if\_cost\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface.
- ← *ifCost* The output cost for the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_COST\_INVALID  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.262** `int smi_ospf_if_cost_set_by_addr (struct smiclient_globals * azg,  
 u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr, u_int32_t  
ifCost)`

Sets This function sets the output cost of the interface of the specific IP address. smi\_ospf\_if\_cost\_set\_by\_addr

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ifName* Name of the interface.  
 ← *ipAddr* The IP address of the interface.  
 ← *ifCost* The output cost for the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_COST\_INVALID  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.263** `int smi_ospf_if_cost_unset (struct smiclient_globals * azg, u_int32_t  
vrId, char * ifName)`

Sets This function resets the cost for the current interface to the default value 10. smi\_ospf\_if\_cost\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ifName* Name of the interface.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED vrId Virtual Router Id



**2.1.2.264** `int smi_ospf_if_cost_unset_by_addr (struct smiclient_globals * azg,  
u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr)`

Sets This function resets the cost for the specified interface to the default value 10.  
smi\_ospf\_if\_cost\_unset\_by\_addr

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface.
- ← *ipAddr* The IP address of the interface.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.265** `int smi_ospf_if_cost_value_unset (struct smiclient_globals * azg,  
u_int32_t vrId, char * ifName, u_int32_t ifCost)`

Sets This function resets the cost for the current interface to the default value 10. smi\_ospf\_if\_cost\_value\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *name* Name of the interface.
- ← *cost* The output cost for the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED vrId Virtual Router Id

**2.1.2.266** `int smi_ospf_if_database_filter_set (struct smiclient_globals * azg,  
u_int32_t vrId, char * ifName)`

This function suppresses all LSA during synchronization and flooding on a particular interface. smi\_ospf\_if\_database\_filter\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.267** `int smi_ospf_if_database_filter_set_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr)`

This function suppresses all LSA during synchronization and flooding for a particular interface by IP address. `smi_ospf_if_database_filter_set_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *ipAddr* IP interface address

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.268** `int smi_ospf_if_database_filter_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

This function restores the forwarding of LSAs. `smi_ospf_if_database_filter_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.269** `int smi_ospf_if_database_filter_unset_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr)`

This function restores the database filter of the interface specified by IP address. `smi_ospf_if_database_filter_unset_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *ipAddr* IP interface address

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.270** `int smi_ospf_if_dead_interval_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_int32_t ifDeadInterval)`

This function sets the router-dead-interval for the current interface. `smi_ospf_if_dead_interval_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifname* Name of the interface
- ← *ifDeadInterval* The interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_DEAD\_INTERVAL\_INVALID  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.271** `int smi_ospf_if_dead_interval_set_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr, u_int32_t ifDeadInterval)`

This function sets the router-dead-interval for the interface specified by the IP address. `smi_ospf_if_dead_interval_set_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *ipAddr* The IP address of the interface
- ← *ifDeadInterval* The interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_DEAD\_INTERVAL\_INVALID  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.272** `int smi_ospf_if_dead_interval_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

This function resets the dead interval of the interface specified by IP address to the default. `smi_ospf_if_dead_interval_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.273** `int smi_ospf_if_dead_interval_unset_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr)`

This function unsets the router-dead-interval for the interface specified by the IP address. `smi_ospf_if_dead_interval_set_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface

← *ipAddr* The IP address of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.274** `int smi_ospf_if_disable_all_set (struct smiclient_globals * azg,  
u_int32_t vrid, char * ifName)`

This function disables all packet processing on a particular interface regardless whether the network area command is configured. smi\_ospf\_if\_disable\_all\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *vrid* Virtual Router Id  
← *ifName* Name of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.275** `int smi_ospf_if_disable_all_unset (struct smiclient_globals * azg,  
u_int32_t vrid, char * ifName)`

This function unconfigures the ip ospf disable all command. smi\_ospf\_if\_disable\_all\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *vrid* Virtual Router Id  
← *ifName* Name of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.276 `int smi_ospf_if_dna_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

Sets flood-reduction. `smi_ospf_if_dna_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *name* Interface name

#### Returns:

0 on OSPF\_API\_SET\_SUCCESS, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.277 `int smi_ospf_if_dna_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

Unsets flood-reduction. `smi_ospf_if_dna_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *name* Interface name

#### Returns:

0 on OSPF\_API\_SET\_SUCCESS, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

`smi_ospf_if_dna_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *name* Interface name

#### Returns:

0 on OSPF\_API\_SET\_SUCCESS, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR \

**2.1.2.278** `int smi_ospf_if_hello_interval_set (struct smiclient_globals * azg,  
u_int32_t vrId, char * ifName, u_int32_t ifHelloInterval)`

Sets This function sets the hello interval for the current interface. smi\_ospf\_if\_hello\_interval\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface.
- ← *ifHelloInterval* The interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_IF\_HELLO\_INTERVAL\_INVALID  
OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.279** `int smi_ospf_if_hello_interval_set_by_addr (struct smiclient_globals  
* azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr,  
u_int32_t ifHelloInterval)`

This function sets the hello interval for the interface specified by IP address. smi\_ospf\_if\_hello\_interval\_set\_by\_addr

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface.
- ← *ipAddr* The IP address of the interface.
- ← *ifHelloInterval* The interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.280** `int smi_ospf_if_hello_interval_unset (struct smiclient_globals * azg,  
u_int32_t vrId, char * ifName)`

This function resets the hello interval of the current interface to the default value. smi\_ospf\_if\_hello\_interval\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.281** `int smi_ospf_if_hello_interval_unset_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr)`

This function resets the hello interval of the interface specified by IP address to the default value. `smi_ospf_if_hello_interval_unset_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *ipAddr* The IP address of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.282** `int smi_ospf_if_ip_router_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr areaId, int areaFormat, int ospfProcessId, int ospfInterfaceInstanceId, char * cmdOptionalString)`

Sets This function enable the OSPF on interface. `smi_ospf_if_ip_router_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name
- ← *areaId* Area id



- ← *areaFormat* Configured area format
- ← *ospfProcessId* OSPF process ID
- ← *ospfInterfaceInstanceId* OSPF instance ID
- ← *cmdOptionalString* Command optional string

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.283** `int smi_ospf_if_ip_router_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr areaId, int areaFormat, int ospfProcessId, int ospfInterfaceInstanceId)`

Sets This function removes OSPF on interface. smi\_ospf\_if\_ip\_router\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name
- ← *area\_id* Area id
- ← *areaFormat* Configured area format
- ← *ospfProcessId* OSPF process ID
- ← *ospfInterfaceInstanceId* OSPF instance ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.284** `int smi_ospf_if_message_digest_key_get (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_char msgDigestKeyId, char * ifAuthKey)`

This function gets the MD5 key for the mentioned interface. smi\_ospf\_if\_message\_digest\_key\_get

**Parameters:**

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

- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *msgDigestKeyId* The key ID.<1-255>
- *ifauthKey* The MD 5 key. This is a null terminated value

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.285** `int smi_ospf_if_message_digest_key_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_char msgDigestKeyId, char * md5, char * ifAuthKey)`

This function sets the MD5 key for the current interface. `smi_ospf_if_message_digest_key_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *msgDigestKeyId* The key ID.<1-255>
- ← *md5* Encryption enabled or not
- ← *ifAuthKey* The MD 5 key. This is a null terminated value

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_MD5\_KEY\_EXIST

OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.286** `int smi_ospf_if_message_digest_key_set_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr, u_char msgDigestKeyId, char * md5, char * ifAuthKey)`

This function sets the MD5 key for the interface specified by IP address. `smi_ospf_if_message_digest_key_set_by_addr`

**Parameters:**

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

- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *ipAddr* The IP address of the interface
- ← *msgDigestKeyId* The key ID.
- ← *md5* Encryption enabled or not
- ← *ifAuthKey* The MD 5 key. This is a null terminated value

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_MD5\_KEY\_EXIST
- OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.287 int smi\_ospf\_if\_message\_digest\_key\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName, u\_char msgDigestKeyId)

This function deletes the MD5 key for the current interface. smi\_ospf\_if\_message\_digest\_key\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *msgDigestKeyId* The key ID.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_MD5\_KEY\_EXIST
- OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

### 2.1.2.288 int smi\_ospf\_if\_message\_digest\_key\_unset\_by\_addr (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName, struct pal\_in4\_addr ipAddr, u\_char msgDigestKeyId)

This function deletes the MD5 key for the interface specified by IP address. smi\_ospf\_if\_message\_digest\_key\_unset\_by\_addr

**Parameters:**

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

- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *ipAddr* The IP address of the interface
- ← *msgDigestKeyId* The key ID.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_MD5\_KEY\_EXIST

OSPF\_API\_SET\_MALLOC\_ERR

OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.289** `int smi_ospf_if_mtu_ignore_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

This function sets ospf not to check mtu size during database description exchange. `smi_ospf_if_mtu_ignore_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.290** `int smi_ospf_if_mtu_ignore_set_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr)`

This function sets ospf not to check mtu size during database description exchange for particular address. `smi_ospf_if_mtu_ignore_set_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name
- ← *ipAddr* Address of the interface.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.291 int smi\_ospf\_if\_mtu\_ignore\_unset (struct smiclient\_globals \* *azg*,  
u\_int32\_t *vrId*, char \* *ifName*)**

This function unconfigurs the ospf mtu ignorance of during database description exchange. smi\_ospf\_if\_mtu\_ignore\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *vrId* Virtual Router Id  
← *ifName* Name of the interface.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.292 int smi\_ospf\_if\_mtu\_ignore\_unset\_by\_addr (struct smiclient\_globals  
\* *azg*, u\_int32\_t *vrId*, char \* *ifName*, struct pal\_in4\_addr *ipAddr*)**

This function unconfigurs the ospf mtu ignorance of during database description exchange for particular address. smi\_ospf\_if\_mtu\_ignore\_unset\_by\_addr

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *vrId* Virtual Router Id  
← *ifname* Name of the interface.  
← *ipAddr* Address of the interface.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.293 `int smi_ospf_if_mtu_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_int16_t ifMtu)`

Sets This function sets mtu size. `smi_ospf_if_mtu_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name
- ← *ifMtu* Mtu size.

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.294 `int smi_ospf_if_mtu_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

Sets This function resets mtu size to default. `smi_ospf_if_mtu_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.295 `int smi_ospf_if_network_p2mp_nbma_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

Sets This function configures an interface to Point-to-Multipoint Non-Broadcast mode. `smi_ospf_if_network_p2mp_nbma_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *name* The interface name

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_MALLOC\_ERR  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.296 int smi\_ospf\_if\_network\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName, int ifNetworkType)**

Sets This function configures the OSPF network type as specified. smi\_ospf\_if\_network\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name
- ← *ifNetworkType* The network type (Point-to-Point | Broadcast | nbma | Point-to-MultiPoint)

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_MALLOC\_ERR  
OSPF\_API\_SET\_ERR\_NETWORK\_TYPE\_INVALID  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.297 int smi\_ospf\_if\_network\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName)**

Sets This function resets the network type to the default type. smi\_ospf\_if\_network\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *name* The interface name

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.298 `int smi_ospf_if_passive_interface_set (struct smiclient_globals * azg, int vrId, char * ifName)`

Sets This function configures the passive interface. `smi_ospf_if_passive_interface_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.299 `int smi_ospf_if_passive_interface_unset (struct smiclient_globals * azg, int vrId, char * ifName)`

Sets This function unconfigure the passive interface. `smi_ospf_if_passive_interface_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.300 `int smi_ospf_if_priority_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_char ifPriority)`

Sets This function sets the priority of the current interface. `smi_ospf_if_priority_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* The interface name



← *ifPriority* The router priority <0-255>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.301** `int smi_ospf_if_priority_set_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr, u_char ifPriority)`

Sets This sets the priority of the interface specified by IP address. smi\_ospf\_if\_priority\_set\_by\_addr

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ifName* The interface name  
 ← *ipAddr* The IP address of the interface  
 ← *ifPriority* The router priority <0-255>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.302** `int smi_ospf_if_priority_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

Sets This function resets the priority of the current interface to the default value 1. smi\_ospf\_if\_priority\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ifName* The interface name

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.303 `int smi_ospf_if_priority_unset_by_addr (struct smiclient_globals * azg, u_int32_t vrid, char * ifName, struct pal_in4_addr ipAddr)`

Sets This function resets the priority of the interface specified. by IP address of the default value 1. `smi_ospf_if_priority_unset_by_addr`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrid* Virtual Router Id
- ← *ifName* The interface name
- ← *ipAddr* The IP address of the interface.

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.304 `int smi_ospf_if_resync_timeout_set (struct smiclient_globals * azg, u_int32_t vrid, char * ifName, u_int32_t ifResyncTimeout)`

This function sets the timeout interval for re-synchronization. If out-of-band re-synchronization does not occur, adjacency is reset. `smi_ospf_if_resync_timeout_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrid* Virtual Router Id
- ← *ifName* Name of the interface
- ← *ifResyncTimeout* The re-synchronization timeout in seconds: <1-65535>

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_IF\_RESYNC\_TIMEOUT\_INVALID

### 2.1.2.305 `int smi_ospf_if_resync_timeout_set_by_addr (struct smiclient_globals * azg, u_int32_t vrid, char * ifName, struct pal_in4_addr ipAddr, u_int32_t ifResyncTimeout)`

This function the time between LSA retransmission for adjacencies belonging to the interface by ip address. `smi_ospf_if_resync_timeout_set_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *ipAddr* The IP address of the interface
- ← *ifResyncTimeout* The re-synchronization timeout in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_IF\_RESYNC\_TIMEOUT\_INVALID

**2.1.2.306** `int smi_ospf_if_resync_timeout_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

This function unsets the timeout interval for re-synchronization. smi\_ospf\_if\_resync\_timeout\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.307** `int smi_ospf_if_resync_timeout_unset_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr)`

This function unsets the timeout interval for re-synchronization for particular address. smi\_ospf\_if\_resync\_timeout\_unset\_by\_addr

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface

← *ipAddr* The IP address of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.308 int smi\_ospf\_if\_retransmit\_interval\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName, u\_int32\_t ifRetransmitInterval)**

Sets This function sets the time between LSA retransmission for adjacencies belonging to the interface. smi\_ospf\_if\_retransmit\_interval\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ifName* Name of the interface.  
 ← *ifRetransmitInterval* The interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.309 int smi\_ospf\_if\_retransmit\_interval\_set\_by\_addr (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName, struct pal\_in4\_addr ipAddr, u\_int32\_t ifRetransmitInterval)**

Sets This function the time between LSA retransmission for adjacencies belonging to the interface by ip address. smi\_ospf\_if\_retransmit\_interval\_set\_by\_addr

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ifName* Name of the interface.  
 ← *ipAddr* The IP address of the interface  
 ← *ifRetransmitInterval* The interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.310 `int smi_ospf_if_retransmit_interval_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

Sets This function resets the retransmit interval of the current interface to the default.  
smi\_ospf\_if\_retransmit\_interval\_unset

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface.

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

### 2.1.2.311 `int smi_ospf_if_retransmit_interval_unset_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr)`

Sets This function resets the retransmit interval of the interface specified by IP address to the default. smi\_ospf\_if\_retransmit\_interval\_unset\_by\_addr

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface.
- ← *ipAddr* address

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

### 2.1.2.312 `int smi_ospf_if_te_metric_set (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, u_int32_t ifTEMetric)`

This function sets the TE-metric on the specified interface. smi\_ospf\_if\_te\_metric\_set

#### Parameters:

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

- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface
- ← *ifTEMetric* TE metric <1-65535>.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_TELINK\_METRIC\_EXIST
- OSPF\_API\_SET\_MALLOC\_ERR
- OSPF\_API\_SET\_ERR\_IF\_COST\_INVALID

### 2.1.2.313 int smi\_ospf\_if\_te\_metric\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName)

This function unsets the TE-metric on a particular interface. smi\_ospf\_if\_te\_metric\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_TELINK\_METRIC\_EXIST
- OSPF\_API\_SET\_ERR\_IF\_COST\_INVALID

### 2.1.2.314 int smi\_ospf\_if\_transmit\_delay\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, char \* ifName, u\_int32\_t ifTransmitDelay)

This function sets the transmit delay interval (seconds) for the current interface. smi\_ospf\_if\_transmit\_delay\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface.
- ← *ifTransmitDelay* The interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.315** `int smi_ospf_if_transmit_delay_set_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr, u_int32_t ifTransmitDelay)`

This function sets the transmit delay interval (seconds) for the interface specified by IP address. `smi_ospf_if_transmit_delay_set_by_addr`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ifName* Name of the interface.  
 ← *ipAddr* The IP address of the interface  
 ← *ifTransmitDelay* The interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.316** `int smi_ospf_if_transmit_delay_unset (struct smiclient_globals * azg, u_int32_t vrId, char * ifName)`

Sets This function resets the transmit delay interval of the current interface. `smi_ospf_if_transmit_delay_unset`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ifName* Name of the interface.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.317** `int smi_ospf_if_transmit_delay_unset_by_addr (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, struct pal_in4_addr ipAddr)`

Sets This function resets the transmit delay interval of the interface specified by IP address to the default. `smi_ospf_if_transmit_delay_unset_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ifName* Name of the interface.
- ← *ipAddr* The IP address of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.318** `int smi_ospf_log_adj_changes_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char * logAdjacencyDebugType)`

Sets This function configure the log adjacency. `smi_ospf_log_adj_changes_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* Ospf process Id
- ← *logAdjacencyDebugType* pointer to debug type bref or detail

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.319** `int smi_ospf_log_adj_changes_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char * logAdjacencyDebugType)`

Sets This function unconfigure the log adjacency. `smi_ospf_log_adj_changes_unset`

**Parameters:**

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



- ← *vrId* Virtual Router Id
- ← *ospfProcessId* Ospf process Id
- ← *str* pointer to debug type bref or detail

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.320 int smi\_ospf\_lsa\_min\_arrival\_set (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId, u\_int32\_t lsaMinArrival)

Sets This function sets the minimum interval to accept the same link-state advertisement (LSA) from OSPF neighbors. smi\_ospf\_lsa\_min\_arrival\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *lsaMinArrival* The minimum delay in milliseconds between accepting the same LSA from neighbors

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.321 int smi\_ospf\_lsa\_min\_arrival\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId)

Sets This function sets the minimum interval to accept the same link-state advertisement (LSA) from OSPF neighbors to its default value (1000 milliseconds). smi\_ospf\_lsa\_min\_arrival\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.322** `int smi_ospf_lsa_throttle_timers_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, u_int32_t throttleTimersStartDelay, u_int32_t holdInterval, u_int32_t throttleTimersMaxDelay)`

Sets This function sets the rate-limiting intervals for OSPF link-state advertisement (LSA) generation. smi\_ospf\_lsa\_throttle\_timers\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure

← *vrId* Virtual Router Id

← *ospfProcessId* OSPF process ID <0-65535>

← *lsaTTStartDelay* generation of LSAs <0-600000>.

← *holdInterval* The hold time in milliseconds <0-600000>.

← *lsaTTMaxDelay* The maximum wait time in milliseconds between generation of the same LSA <0-600000>

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_TIMER\_VALUE\_INVALID

**2.1.2.323** `int smi_ospf_lsa_throttle_timers_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

Sets This function sets the rate-limiting intervals for OSPF link-state advertisement (LSA) generation to their default values. smi\_ospf\_lsa\_throttle\_timers\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure

← *vrId* Virtual Router Id

← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.324** `int smi_ospf_max_area_limit_set_sdkapi (struct smiclient_globals * azg, int ospfProcessId, u_int32_t vrId, u_int32_t maxAreaLimit)`

Sets the maximum number of OSPF areas. smi\_ospf\_max\_area\_limit\_set\_sdkapi

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* proc id
- ← *vrId* vrid
- ← *maxAreaLimit* Maximum number of OSPF areas <14294967294>

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.325** `int smi_ospf_max_area_limit_unset_sdkapi (struct smiclient_globals * azg, int ospfProcessId, u_int32_t vrId)`

Removes the maximum number of OSPF areas that was set. smi\_ospf\_max\_area\_limit\_unset\_sdkapi

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* Process id
- ← *vrId* Virtual Router Id

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.326** `int smi_ospf_max_concurrent_dd_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, u_int16_t maxDBDescriptor)`

Sets This function sets the specified limit for the number of concurrently processed Database Descriptors. smi\_ospf\_max\_concurrent\_dd\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *maxDBDescriptor* Database Descriptor (DD) processes <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.327 `int smi_ospf_max_concurrent_dd_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

Sets This function resets the specified limit for the number of concurrently processed Database Descriptors. `smi_ospf_max_concurrent_dd_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.328 `int smi_ospf_max_unuse_lsa_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, u_int32_t maxLsaPacket)`

This function sets maximum number of link state advertisement packet to specified value. `smi_ospf_max_unuse_lsa_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *maxLsaPacket* no of lsa packets <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.329 int smi\_ospf\_max\_unuse\_lsa\_unset (struct smiclient\_globals \* *azg*,  
u\_int32\_t *vrId*, int *ospfProcessId*)**

Sets This function sets maximum number of link state advertisement packet to default value. smi\_ospf\_max\_unuse\_lsa\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *vrId* Virtual Router Id  
← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.330 int smi\_ospf\_max\_unuse\_packet\_set (struct smiclient\_globals \* *azg*,  
u\_int32\_t *vrId*, int *ospfProcessId*, u\_int32\_t *maxUnusePackets*)**

This function sets the maximum no of unused ospf packets. smi\_ospf\_max\_unuse\_packet\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *vrId* Virtual Router Id  
← *ospfProcessId* OSPF process ID <0-65535>  
← *maxUnusePackets* no of packets unused

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.331 `int smi_ospf_max_unuse_packet_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

This function sets the maximum no of unused ospf packets to default value. `smi_ospf_max_unuse_packet_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.332 `s_int32_t smi_ospf_multi_area_adjacency_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, char * ifName, struct pal_in4_addr mlinkNbr, int areaFormat)`

This call implements the area multi-area-adjacency command to enable the multiple-area adjacency on specified interface for the given area ID. `smi_ospf_multi_area_adjacency_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *ifName* The interface name
- ← *mlinkNbr* The neighbor IP address
- ← *areaFormat* The format of the address: OSPF\_AREA\_ID\_FORMAT\_ADDRESS OSPF\_AREA\_ID\_FORMAT\_DECIMAL

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_LIMIT  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_MULTI\_AREA\_LINK\_CANT\_GET

**2.1.2.333** `s_int32_t smi_ospf_multi_area_adjacency_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, u_char * ifName, struct pal_in4_addr mLinkNbr)`

This call implements the no parameter of the area multi-area-adjacency command to disable multi-areaadjacency. `smi_ospf_multi_area_adjacency_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *ifName* The interface name
- ← *mLinkNbr* The neighbor IP address

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_MULTI\_AREA\_ADJ\_NOT\_SET  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

**2.1.2.334** `int smi_ospf_nbr_static_cost_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr neighborIpAddress, u_int16_t neighborCost)`

This function sets the cost of the specified non-broadcast multi-access (nbma) neighbor. `smi_ospf_nbr_static_cost_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *neighborIpAddress* The nbma neighbor IP address
- ← *neighborCost* The link state metric to this neighbor <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_NBR\_STATIC\_NOT\_EXIST

**2.1.2.335** `int smi_ospf_nbr_static_cost_unset (struct smiclient_globals *  
azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr  
neighborIpAddr)`

This function resets the cost of the specified non-broadcast multi-access (nbma) neighbor to the default value 0. `smi_ospf_nbr_static_cost_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *neighborIpAddr* The nbma neighbor IP address

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_NBR\_STATIC\_NOT\_EXIST

**2.1.2.336** `int smi_ospf_nbr_static_poll_interval_set (struct smiclient_globals  
* azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr  
neighborIpAddr, int nbrPollInterval)`

This function sets the poll interval of the non-broadcast multi-access (nbma) neighbor. `smi_ospf_nbr_static_poll_interval_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *neighborIpAddr* The nbma neighbor IP address
- ← *nbrPollInterval* The poll interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_NBR\_STATIC\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_NBR\_P2MP\_CONFIG\_INVALID



**2.1.2.337** `int smi_ospf_nbr_static_poll_interval_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr neighborIpAddr)`

This function resets the poll interval of the specified non-broadcast multi-access (nbma) neighbor to the default value. `smi_ospf_nbr_static_poll_interval_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *neighborIpAddr* The nbma neighbor IP address

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_NBR\_STATIC\_NOT\_EXIST

**2.1.2.338** `int smi_ospf_nbr_static_priority_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr neighborIpAddr, u_char neighborStaticPriority)`

This function sets the priority of the specified non-broadcast multi-access (nbma) neighbor. `smi_ospf_nbr_static_priority_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *neighborIpAddr* The nbma neighbor IP address
- ← *nbr\_static\_priority* The neighbor priority <0-255>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_NBR\_STATIC\_NOT\_EXIST

**2.1.2.339** `int smi_ospf_nbr_static_priority_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr neighborIpAddr)`

This function resets the priority of the non-broadcast multi-access (nbma) neighbor to the default value. `smi_ospf_nbr_static_priority_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *neighborIpAddr* The nbma neighbor IP address

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_NBR\_STATIC\_NOT\_EXIST

**2.1.2.340** `int smi_ospf_nbr_static_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr neighborIpAddr)`

This function sets the non-broadcast multi-access (nbma) neighbor. `smi_ospf_nbr_static_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *neighborIpAddr* The nbma neighbor IP address

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_NBR\_STATIC\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.341** `int smi_ospf_nbr_static_unset (struct smiclient_globals *  
azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr  
neighborIpAddr)`

This function deletes the static non-broadcast multi-access (nbma) neighbor. smi\_ospf\_nbr\_static\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *neighborIpAddr* The nbma neighbor IP address

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_NBR\_STATIC\_NOT\_EXIST

**2.1.2.342** `int smi_ospf_network_set (struct smiclient_globals * azg, u_int32_t  
vrId, int ospfProcessId, struct pal_in4_addr ospfNetAddr,  
u_char netMask, struct pal_in4_addr areaId, s_int16_t  
ospfInterfaceInstanceId)`

Sets This function enables an interface for the OSPF domain. smi\_ospf\_network\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ospfNetAddr* nbma neighbor IP address
- ← *netMask* prefix length
- ← *areaId* area to which the network belongs.
- ← *ospfInterfaceInstanceId* interface instance ID <0-255>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

```

OSPF_API_SET_ERR_IF_INST_ID_CANT_SET
OSPF_API_SET_ERR_IF_INSTANCE_ID_INVALID
OSPF_API_SET_ERR_NETWORK_OWNED_BY_ANOTHER_AREA
OSPF_API_SET_ERR_NETWORK_WITH_ANOTHER_INST_ID_EXIST

```

**2.1.2.343** `int smi_ospf_network_unset (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, struct pal_in4_addr ospfNetAddr,  
u_char netMask, struct pal_in4_addr areaId, s_int16_t  
ospfInterfaceInstanceId)`

Sets This function deletes the network area configuration directive matched to a specified prefix and area. `smi_ospf_network_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ospfNetAddr* nbma neighbor IP address
- ← *netMask* prefix length
- ← *areaId* area to which the network belongs.
- ← *ospfInterfaceInstanceId* interface instance ID <0-255>

**Returns:**

```

OSPF_API_SET_SUCCESS on success, otherwise one of the following error
codes
OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_ERR_PROCESS_ID_INVALID
OSPF_API_SET_ERR_PROCESS_NOT_EXIST
OSPF_API_SET_ERR_NETWORK_NOT_EXIST
OSPF_API_SET_ERR_IF_INSTANCE_ID_INVALID
OSPF_API_SET_ERR_AREA_ID_NOT_MATCH
OSPF_API_SET_ERR_IF_INST_ID_NOT_MATCH

```

**2.1.2.344** `int smi_ospf_opaque_area_lsa_set (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId,  
u_char opaqueType, u_int32_t opaqueId, char * opaqueData,  
u_int32_t opaqueLen)`

This function generates area Opaque LSAs. `smi_ospf_opaque_area_lsa_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id

- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *opaqueType* Opaque type
- ← *opaqueId* Opaque id
- ← *opaqueData* Opaque data
- ← *opaqueLen* Length of Opaque data

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST

**2.1.2.345** `int smi_ospf_opaque_as_lsa_set (struct smiclient_globals * azg,  
 u_int32_t vrId, int ospfProcessId, u_char opaqueType, u_int32_t  
 opaqueId, char * opaqueData, u_int32_t opaqueLen)`

This function generates Autonomous System (AS) area opaque LSAs. `smi_ospf_opaque_as_lsa_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *opaqueType* Opaque type
- ← *opaqueId* Opaque id
- ← *opaqueData* Opaque data
- ← *opaqueLen* Length of Opaque data

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.346** `int smi_ospf_opaque_link_lsa_set (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, struct pal_in4_addr ipAddr,  
u_char opaqueType, u_int32_t opaqueId, char * opaqueData,  
u_int32_t opaqueLen)`

This function generates the specified AS-wide Opaque LSA. `smi_ospf_opaque_link_lsa_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *opaqueType* Opaque type
- ← *opaqueId* Opaque id
- ← *opaqueData* Opaque data
- ← *opaqueLen* Length of Opaque data

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_NOT\_EXIST

**2.1.2.347** `int smi_ospf_opaque_te_link_local_lsa_disable (struct  
smiclient_globals * azg, u_int32_t vrId, char * TELinkName)`

This function disables the exchange of TE link local LSA for the specified TE link. `smi_ospf_opaque_te_link_local_lsa_disable`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *TELinkName* The TE link name

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_ABR\_TYPE\_INVALID

### 2.1.2.348 `int smi_ospf_opaque_te_link_local_lsa_enable (struct smiclient_globals * azg, u_int32_t vrId, char * TELinkName)`

This function enables the exchange of TE link local LSA for a specified GMPLS TE link and determines the remote interface ID of the TE link. `smi_ospf_opaque_te_link_local_lsa_enable`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *TELinkName* The TE link name

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_ABR\_TYPE\_INVALID

### 2.1.2.349 `int smi_ospf_overflow_database_external_interval_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, int exitOverflowInterval)`

Sets This function sets the value of the time-to-recover interval of the overflow state. `smi_ospf_overflow_database_external_interval_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *interval* The time-to-recover interval in seconds.

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.350 `int smi_ospf_overflow_database_external_interval_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

Sets This function resets the value of the interval of the overflow state as defined by `OSPF_DEFAULT_EXIT_OVERFLOW_INTERVAL`. `smi_ospf_overflow_database_external_interval_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.351** `int smi_ospf_overflow_database_external_limit_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, u_int32_t lsdBExternalLimit)`

Sets This function sets the maximum number of LSAs as specified. `smi_ospf_overflow_database_external_limit_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *lsdBExternalLimit* The limit <0-2147483647>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.352** `int smi_ospf_overflow_database_external_limit_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

This call implements the no parameter of the enable ext-ospf-multi-inst command to disable support of multiple OSPF instances on a subnet. `smi_ospf_overflow_database_external_limit_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id



**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.353 int smi\_ospf\_passive\_interface\_default\_set (struct smiclient\_globals \* azg, u\_int32\_t vrid, int ospfProcessId)**

Sets This function sets all interfaces to passive mode by default. smi\_ospf\_passive\_interface\_default\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *vrid* Virtual Router Id  
← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.354 int smi\_ospf\_passive\_interface\_default\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrid, int ospfProcessId)**

Sets This function resets all interfaces. smi\_ospf\_passive\_interface\_default\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
← *vrid* Virtual Router Id  
← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.355** `int smi_ospf_passive_interface_set (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, char * passiveIfName)`

Sets This function sets the specified interface to passive mode (OSPF\_IF\_PASSIVE).  
smi\_ospf\_passive\_interface\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *passiveIfName* The interface name.

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.356** `int smi_ospf_passive_interface_set_by_addr (struct smiclient_globals  
* azg, u_int32_t vrId, int ospfProcessId, char * passiveIfName, struct  
pal_in4_addr passiveIfIpAddr)`

Sets This function sets the interface specified by IP address to passive mode. smi\_ospf\_passive\_interface\_set\_by\_addr

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ifName* The interface name.
- ← *passiveIfIpAddr* The IP address of the interface.

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.357** `int smi_ospf_passive_interface_unset (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, char * ifName)`

Sets This function resets the current interface to active mode. `smi_ospf_passive_interface_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *passiveIfName* The interface name.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.358** `int smi_ospf_passive_interface_unset_by_addr (struct  
smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char *  
passiveIfName, struct pal_in4_addr passiveIfIpAddr)`

Sets This function resets the current interface to active mode. `smi_ospf_passive_interface_unset_by_addr`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *passiveIfName* The interface name.
- ← *passiveIfIpAddr* The IP address of the interface

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.359 `int smi_ospf_process_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

Sets This function creates an OSPF instance. `smi_ospf_process_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

#### Returns:

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.360 `int smi_ospf_process_set_vrf (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, char * vrfName)`

Sets VRF name to an OSPF process. `smi_ospf_process_set_vrf`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *vrfName* VRF name to be set

#### Returns:

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_DOES\_NOT\_EXISTS  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_VRF\_DOES\_NOT\_EXISTS  
 OSPF\_API\_SET\_ERR\_VRF\_ALREADY\_BOUND  
 OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.361 `int smi_ospf_process_shut_set (struct smiclient_globals * azg, u_int32_t vrId, u_int32_t ospfProcessId)`

This function configures the ip ospf shutdown command. `smi_ospf_process_shut_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospf\_id* OSPF process ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.362 **int smi\_ospf\_process\_shut\_unset (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, u\_int32\_t *ospfProcessId*)**

This function unconfigures the ip ospf shutdown command. smi\_ospf\_process\_shut\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospf\_id* OSPF process ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

### 2.1.2.363 **int smi\_ospf\_process\_unset (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *ospfProcessId*)**

Sets This function destroys the specified OSPF process. smi\_ospf\_process\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

### 2.1.2.364 `int smi_ospf_redist_default_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, int routeSourceOrigin)`

Sets OSPF to redistribute default route into an OSPF instance.

When set, the router acts like an ASBR to redistribute routes into an OSPF instance.

By default an ASBR does not generate a default route into an OSPF instance. `smi_ospf_redist_default_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceOrigin* Source for default route originated <1-2> 1-NSM:  
2-Always: Advertise always regardless of whether there is a default route or not

#### Returns:

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_DEFAULT\_ORIGIN\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

### 2.1.2.365 `int smi_ospf_redist_default_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

Sets OSPF not to redistribute default route into an OSPF instance. `smi_ospf_redist_default_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>

#### Returns:

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.366** `int smi_ospf_redist_metric_set (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, int routeSourceType, int  
redistMetricValue, int secondaryOspfProcessId)`

Sets OSPF to redistribute routes with an external metric value. `smi_ospf_redist_metric_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceType* Source for the route (0|1|2|3|4|6|8|9)  
0-Default, 1-Kernel, 2-Connected, 3-Static,  
4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *redistMetricValue* Metric value <0-16777214>
- ← *secondaryOspfProcessId* Another OSPF instance ID <0-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_METRIC\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.367** `int smi_ospf_redist_metric_type_set (struct smiclient_globals *  
azg, u_int32_t vrId, int ospfProcessId, int routeSourceType, int  
redistMetricType, int secondaryOspfProcessId)`

Sets OSPF to redistribute routes with an external metric type. `smi_ospf_redist_metric_type_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceType* Source for the route (0|1|2|3|4|6|8|9)  
0-Default, 1-Kernel, 2-Connected, 3-Static,  
4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *redistMetricType* Metric type <0-2>  
0-Unspecified, 1-Type1, 2-Type2
- ← *secondaryOspfProcessId* Another OSPF instance ID <0-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_METRIC\_TYPE\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.368** `int smi_ospf_redist_metric_unset (struct smiclient_globals *  
 azg, u_int32_t vrId, int ospfProcessId, int routeSourceType, int  
 secondaryOspfProcessId)`

Sets OSPF not to redistribute routes with an external metric type. `smi_ospf_redist_metric_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceType* Source for the route (0|1|2|3|4|6|8|9)  
 0-Default, 1-Kernel, 2-Connected, 3-Static,  
 4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *secondaryOspfProcessId* Another OSPF instance ID <0-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.369** `int smi_ospf_redist_metric_unset (struct smiclient_globals *  
 azg, u_int32_t vrId, int ospfProcessId, int routeSourceType, int  
 secondaryOspfProcessId)`

Sets OSPF not to redistribute routes with an external metric value. `smi_ospf_redist_metric_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>



- ← *routeSourceType* Source for the route (0|1|2|3|4|6|8|9)  
0-Default, 1-Kernel, 2-Connected, 3-Static,  
4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *secondaryOspfProcessId* Another OSPF instance ID <0-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.370** `int smi_ospf_redist_proto_set (struct smiclient_globals * azg,  
 u_int32_t vrId, int ospfProcessId, int routeSourceType, int  
 secondaryOspfProcessId)`

Sets OSPF to redistribute connected/kernel/static routes and routes from different routing protocols into another OSPF instance. smi\_ospf\_redist\_proto\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceType* Source for the route (1|2|3|4|6|8|9)  
1-Kernel, 2-Connected, 3-Static,  
4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *secondaryOspfProcessId* Another OSPF instance ID <1-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_SELF\_REDIST

**2.1.2.371** `int smi_ospf_redist_proto_unset (struct smiclient_globals *  
 azg, u_int32_t vrId, int ospfProcessId, int routeSourceType, int  
 secondaryOspfProcessId)`

Sets OSPF not to redistribute connected/kernel/static routes and routes from different routing protocols into another OSPF instance. smi\_ospf\_redist\_proto\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceType* Source for the route (1|2|3|4|6|8|9)
  - 1-Kernel, 2-Connected, 3-Static,
  - 4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *secondaryOspfProcessId* Another OSPF instance ID <1-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.372** `int smi_ospf_redist_tag_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, int routeSourceType, u_int32_t redistTag, int secondaryOspfProcessId)`

Sets OSPF to redistribute routes with a tag value to be used as a match for controlling redistribution via route maps. `smi_ospf_redist_tag_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceType* Source for the route (1|2|3|4|6|8|9)
  - 1-Kernel, 2-Connected, 3-Static,
  - 4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *tag* Tag value <0-4294967295>
- ← *ospf\_sospfProcessId* Another OSPF instance ID <1-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.373** `int smi_ospf_redist_tag_unset (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, int routeSourceType, int  
secondaryOspfProcessId)`

Sets OSPF not to redistribute routes with a tag value that was set to use as a match for controlling redistribution via route maps. `smi_ospf_redist_tag_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceType* Source for the route (1|2|3|4|6|8|9)  
1-Kernel, 2-Connected, 3-Static,  
4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *secondaryOspfProcessId* Another OSPF instance ID <1-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.374** `int smi_ospf_redistribute_default_set (struct smiclient_globals *  
azg, u_int32_t vrId, int ospfProcessId, int routeSourceOrigin, int  
redistMetricType, int redistMetricValue)`

Sets OSPF to redistribute default routes into another OSPF instance. `smi_ospf_redistribute_default_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceOrigin* Source for default route originated <0-2> 0-Unspecified  
1-NSM:  
2-Always: Advertise always regardless of whether there is a default route or not
- ← *redistMetricType* Unused variable
- ← *redistMetricValue* Unused variable

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_DEFAULT\_ORIGIN\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.375** `int smi_ospf_redistribute_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, int secondaryOspfProcessId, int routeSourceType, int redistMetricType, int redistMetricValue)`

Sets OSPF to redistribute routes learned from different sources into another OSPF instance. `smi_ospf_redistribute_set`

**Parameters:**

- ← **azg** Pointer to the SMI client global structure
- ← **vrId** Virtual Router ID
- ← **ospfProcessId** OSPF process ID <0-65535>
- ← **secondaryOspfProcessId** Another OSPF instance ID <0-65535>
- ← **routeSourceType** Source for the route <0-10>  
0-Default, 1-Kernel, 2-Connected, 3-Static,  
4-RIP, 5-RIPng, 6-OSPF, 7-OSPF6,  
8-BGP, 9-ISIS, 10-TRILL
- ← **redistMetricType** Metric type <0-2>  
0-Unspecified, 1-Type1, 2-Type2
- ← **redistMetricValue** Metric value <0-16777214>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_METRIC\_TYPE\_INVALID  
 OSPF\_API\_SET\_ERR\_METRIC\_INVALID

**2.1.2.376** `int smi_ospf_restart_graceful_sdkapi (struct smiclient_globals * azg, int restartSeconds, int restartReason, u_int32_t vrId)`

Sets the grace period in seconds for OSPF to restart gracefully along with reason for restart. `smi_ospf_restart_graceful_sdkapi`

**Parameters:**

- ← **azg** Pointer to the SMI client global structure
- ← **grace\_period** Grace period in seconds <1-1800>
- ← **reason** Reason for restart (Unknown, Restart, Upgrade and Redundant Switch)
- ← **vrId** Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes

```

OSPF_API_SET_ERR_GRACE_PERIOD_INVALID
OSPF_API_SET_ERR_INVALID_REASON
OSPF_API_SET_MALLOC_ERR
OSPF_API_SET_ERR_PROCESS_NOT_EXIST
OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_ERR_PROCESS_ID_INVALID

```

### 2.1.2.377 int smi\_ospf\_restart\_helper\_grace\_period\_set (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, int *restartHelperPeriod*)

Sets the maximum grace period allowed to be as helper router for restarting a router. smi\_ospf\_restart\_helper\_grace\_period\_set

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *restartHelperPeriod* Maximum grace period in seconds

#### Returns:

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_GRACE\_PERIOD\_INVALID

### 2.1.2.378 int smi\_ospf\_restart\_helper\_grace\_period\_unset (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*)

Removes the maximum grace period allowed that was set to be as helper router for restarting a router. smi\_ospf\_restart\_helper\_grace\_period\_unset

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID

#### Returns:

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.379 int smi\_ospf\_restart\_helper\_never\_router\_set (struct smiclient\_globals \* *azg*, u\_int32\_t *vrId*, struct pal\_in4\_addr *nbrRouterId*)

Sets the particular neighbor ID to never act as helper. smi\_ospf\_restart\_helper\_never\_router\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *nbrRouterId* Neighbor router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IP\_ADDR\_IN\_USE  
 OSPF\_API\_SET\_ERR\_MALLOC\_FAIL\_FOR\_ROUTERID

**2.1.2.380** `int smi_ospf_restart_helper_never_router_unset (struct smiclient_globals * azg, u_int32_t vrId, struct pal_in4_addr nbrRouterId)`

Unsets the particular neighbor ID that was to never act as helper. `smi_ospf_restart_helper_never_router_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *nbrRouterId* Neighbor router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_NEVER\_RTR\_ID\_NOT\_EXIST

**2.1.2.381** `int smi_ospf_restart_helper_never_router_unset_all (struct smiclient_globals * azg, u_int32_t vrId)`

Removes all neighbor IDs from the router helper list. `smi_ospf_restart_helper_never_router_unset_all`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_EMPTY\_NEVER\_RTR\_ID

**2.1.2.382** `int smi_ospf_restart_helper_policy_set (struct smiclient_globals *  
azg, u_int32_t vrId, int restartHelperPolicy)`

Sets the helper behavior for OSPF graceful restart. `smi_ospf_restart_helper_policy_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *restartHelperPolicy* Helper policy (Never, Only reload, Only Upgrade)

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_INVALID\_HELPER\_POLICY

**2.1.2.383** `int smi_ospf_restart_helper_policy_unset (struct smiclient_globals *  
azg, u_int32_t vrId)`

Resets the helper behavior to default (i.e., always accept). `smi_ospf_restart_helper_policy_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID

**Returns:**

- 0 on success, otherwise one of the following error codes
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.384** `int smi_ospf_routemap_default_set (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, char * routeMapName)`

Sets the default route-map reference for OSPF to redistribute routes via. `smi_ospf_routemap_default_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeMapName* Route map reference name

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.385 int smi\_ospf\_routemap\_default\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrid, int ospfProcessId)**

Unsets the default route-map reference for OSPF to redistribute routes via. smi\_ospf\_routemap\_default\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrid* Virtual Router ID  
 ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.386 int smi\_ospf\_routemap\_set (struct smiclient\_globals \* azg, u\_int32\_t vrid, int ospfProcessId, int routeSourceType, char \* routeMapName, int secondaryOspfProcessId)**

Sets OSPF to redistribute routes via an route-map reference.

When set, OSPF does not look for default network to redistribute routes. smi\_ospf\_routemap\_set

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrid* Virtual Router ID  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *routeSourceType* Source for the route (1|2|3|4|6|8|9)  
     1-Kernel, 2-Connected, 3-Static,  
     4-RIP, 6-OSPF, 8-BGP, 9-ISIS  
 ← *routeMapName* Route map reference name  
 ← *secondaryOspfProcessId* Another OSPF instance ID <1-65535>



**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_NOT\_SET

**2.1.2.387** `int smi_ospf_routemap_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, int routeSourceType, int secondaryOspfProcessId)`

Sets OSPF not to redistribute routes via an route-map reference.

When unset, OSPF look for default network to redistribute routes. `smi_ospf_routemap_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router ID
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routeSourceType* Source for the route (0|1|2|3|4|6|8|9)  
 0-Default, 1-Kernel, 2-Connected, 3-Static,  
 4-RIP, 6-OSPF, 8-BGP, 9-ISIS
- ← *secondaryOspfProcessId* Another OSPF instance ID <0-65535>

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_REDISTRIBUTE\_PROTO\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.1.2.388** `int smi_ospf_router_id_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr routerId)`

Sets This function sets the static OSPF router ID to the specified value. `smi_ospf_router_id_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id

- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *routerId* The router identifier.

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.389 **int smi\_ospf\_router\_id\_unset (struct smiclient\_globals \* azg, u\_int32\_t vrId, int ospfProcessId)**

Sets This function resets the static OSPF router ID to the default value: 0. smi\_ospf\_router\_id\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.1.2.390 **int smi\_ospf\_set\_area\_aggregate\_effect (struct smiclient\_globals \* azg, int ospfProcessId, struct pal\_in4\_addr areaId, int addrAggrType, struct pal\_in4\_addr ipAddr, struct pal\_in4\_addr mask, int areaAggrEffect, u\_int32\_t vrId)**

Sets and object that tells, subnets subsumed by ranges either trigger the advertisement of the indicated aggregate (advertiseMatching) or result in the subnet's not being advertised at all outside the area. smi\_ospf\_set\_area\_aggregate\_effect

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *areaId* Aggregate Area ID
- ← *addrAggrType* Address aggregate type.
- ← *ipAddr* IP address of the Net or Subnet

- ← *mask* Subnet mask
- *areaAggrEffect* Management object for subnets
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE

**2.1.2.391** `int smi_ospf_set_area_aggregate_route_tag (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int addrAggrType, struct pal_in4_addr ipAddr, struct pal_in4_addr mask, u_int32_t routeTag, u_int32_t vrId)`

Sets and object that tells, subnets subsumed by ranges either trigger the advertisement of the indicated aggregate (advertiseMatching) or result in the subnet's not being advertised at all outside the area. smi\_ospf\_set\_area\_aggregate\_route\_tag

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *areaId* Aggregate Area ID
- ← *addrAggrType* Address aggregate type.
- ← *ipAddr* IP address of the Net or Subnet
- ← *mask* Subnet mask
- 
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE

**2.1.2.392** `int smi_ospf_set_area_aggregate_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int addrAggrType, struct pal_in4_addr ipAddr, struct pal_in4_addr mask, int areaAggrStatus, u_int32_t vrId)`

Sets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction. smi\_ospf\_set\_area\_aggregate\_status

**Parameters:**

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

- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *areaId* Aggregate Area ID
- ← *addrAggrType* Address aggregate type.
- ← *ipAddr* IP address of the Net or Subnet
- ← *mask* Subnet mask
- ← *areaAggrStatus* Table management action status
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_INCONSISTENT\_VALUE  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE

**2.1.2.393** `int smi_ospf_set_area_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int areaStatus, u_int32_t vrId)`

Sets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction. `smi_ospf_set_area_status`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- ← *areaStatus* Table management action status (create/construct/destroy)
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_INCONSISTENT\_VALUE  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE  
 OSPF\_API\_GET\_ERROR

**2.1.2.394** `int smi_ospf_set_asbdr_rtr_status (struct smiclient_globals * azg, int ospfProcessId, int areaASBDRRouterStatus, u_int32_t vrId)`

Sets a flag to note whether this router is configured as an Autonomous System Border Router. `smi_ospf_set_asbdr_rtr_status`

**Parameters:**

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

- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaASBDRRouterStatus* Enabled/disabled
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE  
 OSPF\_API\_SET\_ERROR

**2.1.2.395** `int smi_ospf_set_if_admin_stat (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int ifAdminStat, u_int32_t vrId)`

Sets the administrative status of the OSPF interface. The value formed on the interface, and the interface will be advertised as an internal route to some area. The value 'disabled' denotes that the interface is external to OSPF. smi\_ospf\_set\_if\_admin\_stat

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- ← *ifAdminStat* OSPF interface administrative status
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERROR  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE

**2.1.2.396** `int smi_ospf_set_if_area_id (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, struct pal_in4_addr areaId, u_int32_t vrId)`

Sets the 32-bit integer uniquely identifying the area to which the interface connects. Area ID 0.0.0.0 is used for the OSPF backbone. smi\_ospf\_set\_if\_area\_id

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface

- ← *ifIndex* Interface index or zero
- ← *ifAreaId* Area id
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_AREA\_ID\_FORMAT\_INVALID  
 OSPF\_API\_SET\_ERROR

**2.1.2.397** `int smi_ospf_set_if_auth_key (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int authKeyLength, char * ifAuthKey, u_int32_t vrId)`

Sets the OSPF authentication key. `smi_ospf_set_if_auth_key`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- ← *authKeyLength* The length of the key <0-256>
- ← *ifAuthKey* The authentication key
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.398** `int smi_ospf_set_if_auth_type (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int ifAuthType, u_int32_t vrId)`

Sets the authentication type specified for an interface. `smi_ospf_set_if_auth_type`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface

- ← *ifIndex* Interface index or zero
- ← *ifAuthType* OSPF authentication type <0-2> none (0),  
simplePassword (1),  
md5 (2)
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_GET\_ERROR
- OSPF\_API\_SET\_ERR\_WRONG\_VALUE
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_MALLOC\_ERR
- OSPF\_API\_SET\_ERR\_AUTH\_TYPE\_INVALID

**2.1.2.399** `int smi_ospf_set_if_hello_interval (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int  
ifHelloInterval, u_int32_t vrId)`

Sets the interval, in seconds, between the Hello packets that the router sends on the interface. This value must be the same for all routers attached to a common network. smi\_ospf\_set\_if\_hello\_interval Sets the dead interval, in seconds, that a router's Hello packets have not been seen before its neighbors declare the router down. This should be some multiple of the Hello interval. This value must be the same for all routers attached to a common network.

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipaddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- ← *ifHelloInterval* Hello interval
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERROR
- OSPF\_API\_SET\_ERR\_WRONG\_VALUE
- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_MALLOC\_ERR
- OSPF\_API\_SET\_ERR\_IF\_HELLO\_INTERVAL\_INVALID

**2.1.2.400** `int smi_ospf_set_if_metric_status (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int  
typeOfService, int ifMetricStatus, u_int32_t vrId)`

Sets object permits management of the table by facilitating actions such as row creation, construction, and destruction. `smi_ospf_set_if_metric_status`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ipAddr* IP address of this OSPF interface.
- ← *ifIndex* Interface index or zero.
- ← *typeOfService* Type of Service.
- ← *ifMetricStatus* Table management action status.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_GET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERROR  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE  
 OSPF\_API\_SET\_ERR\_INVALID\_VALUE

**2.1.2.401** `int smi_ospf_set_if_metric_value (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int  
typeOfService, int ifMetricValue, u_int32_t vrId)`

Sets the metric of using this Type of Service on this interface. `smi_ospf_set_if_metric_value`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *ipAddr* IP address of this OSPF interface.
- ← *ifIndex* Interface index or zero.
- ← *typeOfService* Type of Service.
- ← *ifMetricValue* Metric value
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERROR



**2.1.2.402** `int smi_ospf_set_if_poll_interval (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int  
ifPollInterval, u_int32_t vrId)`

Sets the interval, in seconds, between the Hello packets sent to an inactive non-broadcast multi-access neighbor. `smi_ospf_set_if_poll_interval`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- ← *ifPollInterval* Polling interval
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERROR  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_NBR\_STATIC\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_NBR\_P2MP\_CONFIG\_INVALID

**2.1.2.403** `int smi_ospf_set_if_retrans_interval (struct smiclient_globals * azg,  
int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int  
ifRetransmitInterval, u_int32_t vrId)`

Sets the interval, in seconds, between link-state-advertisement retransmissions, for adjacencies belonging to this interface. This value is also used when retransmitting database description and Link State request packets. Note that minimal value SHOULD be 1 second. `smi_ospf_set_if_retrans_interval`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- ← *ifRetransmitInterval* Retransmission interval
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

```

OSPF_API_SET_ERROR
OSPF_API_SET_ERR_WRONG_VALUE
OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_MALLOC_ERR

```

**2.1.2.404** `int smi_ospf_set_if_rtr_dead_interval (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int ifDeadInterval, u_int32_t vrid)`

Sets the dead interval, in seconds, that a router's Hello packets have not been seen before its neighbors declare the router down. This should be some multiple of the Hello interval. This value must be the same for all routers attached to a common network. smi\_ospf\_set\_if\_rtr\_dead\_interval

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- ← *ifDeadInterval* Dead interval
- ← *vrid* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

```

OSPF_API_SET_ERROR
OSPF_API_SET_ERR_WRONG_VALUE
OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_MALLOC_ERR
OSPF_API_SET_ERR_IF_DEAD_INTERVAL_INVALID

```

**2.1.2.405** `int smi_ospf_set_if_rtr_priority (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int ifRouterPriority, u_int32_t vrid)`

Sets the priority of this interface. Used in multi-access networks, this field is used in the designated router election algorithm. The value 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, routers will use their Router ID as a tie breaker. smi\_ospf\_set\_if\_rtr\_priority

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>

- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- ← *ifRouterPriority* OSPF interface priority
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERROR

OSPF\_API\_SET\_ERR\_WRONG\_VALUE

**2.1.2.406** `int smi_ospf_set_if_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int ifStatus, u_int32_t vrId)`

Sets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction. smi\_ospf\_set\_if\_status

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- ← *ifStatus* Table management action status
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERROR

OSPF\_API\_SET\_ERR\_WRONG\_VALUE

**2.1.2.407** `int smi_ospf_set_if_transit_delay (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int ifTransmitDelay, u_int32_t vrId)`

Sets the transit-delay value of the OSPF interface, which is an estimate of the number of seconds required to transmit a link-state update packet through this interface. Note that the minimal value SHOULD be 1 second. smi\_ospf\_set\_if\_transit\_delay

**Parameters:**

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

- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *ipAddr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- ← *ifTransmitDelay* Transit delay
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERROR

OSPF\_API\_SET\_ERR\_WRONG\_VALUE

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.408** `int smi_ospf_set_if_type (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr ipAddr, int ifIndex, int ifType, u_int32_t vrId)`

Sets the OSPF interface type. By way of a default, this field may be intuited from the corresponding value of ifType. Broadcast LANs, such as Ethernet and IEEE 802.5, take the value 'broadcast', X.25 and similar technologies take the value 'nbma', and links that are definitively point to point take the value 'pointToPoint'. smi\_ospf\_set\_if\_type

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *addr* IP address of this OSPF interface
- ← *ifIndex* Interface index or zero
- ← *type* Interface type broadcast (1),  
nbma (2),  
pointToPoint (3),  
pointToMultipoint (5)
- ← *vrId* Virtual Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERROR

OSPF\_API\_SET\_ERR\_READONLY

**2.1.2.409** `int smi_ospf_set_import_as_extern (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, int  
areaType)`

Sets the OSPF area type that indicates if an area is a stub area, NSSA, or standard area. Type-5 AS-external LSAs and type-11 Opaque LSAs are not imported into stub areas or NSSAs. NSSAs import AS-external data as type-7 LSAs. `smi_ospf_set_import_as_extern`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* Area ID
- ← *areaType* Area type
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_IS\_NSSA  
 OSPF\_API\_SET\_ERR\_AREA\_IS\_STUB  
 OSPF\_API\_SET\_ERR\_AREA\_LIMIT  
 OSPF\_API\_SET\_ERR\_AREA\_HAS\_VLINK  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE  
 OSPF\_API\_GET\_ERROR

**2.1.2.410** `int smi_ospf_set_lsdb_limit_sdkapi (struct smiclient_globals * azg,  
int ospfProcessId, u_int32_t vrId, u_int32_t lsdbLimit, int actionType,  
int isLsdbLimit)`

Sets the the maximum number of LSAs that can be supported by the OSPF instance, along with action (Hard/Soft) to performed in case the number of LSAs exceeds the specified limit. `smi_ospf_set_lsdb_limit_sdkapi`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *top* Pointer to structure ospf
- ← *lsdblimit* Number of LSAs
- ← *type* Action type (Hard | Soft) Hard: Shutdown the OSPF instance if the number of LSAs exceeds the specified limit.  
 Soft: Send warning to the OSPF instance if the number of LSAs exceeds the specified limit.

← *set* Set to specified limit to zero (Yes/No)

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.411 int smi\_ospf\_set\_multicast\_extensions (struct smiclient\_globals \* azg, int ospfProcessId, int multicastExtStatus, u\_int32\_t vrId)**

Sets a bit mask indicating whether the router is forwarding IP multicast (Class D) datagrams based on the algorithms defined in the multicast extensions to OSPF.

Bit 0, if set, indicates that the router can forward IP multicast datagrams in the router's directly attached areas (called intra-area multicast routing).

Bit 1, if set, indicates that the router can forward IP multicast datagrams between OSPF areas (called inter-area multicast routing).

Bit 2, if set, indicates that the router can forward IP multicast datagrams between Autonomous Systems (called inter-AS multicast routing).

Only certain combinations of bit settings are allowed, namely: 0 (no multicast forwarding is enabled), 1 (intra-area multicasting only), 3 (intra-area and inter-area multicasting), 5 (intra-area and inter-AS multicasting), and 7 (multicasting everywhere). By default, no multicast forwarding is enabled. smi\_ospf\_set\_multicast\_extensions

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *status* Bit mask  
 ← *vrId* Virtual Router ID

**Note:**

Currently this is not supported

**Returns:**

0 on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERROR

**2.1.2.412 int smi\_ospf\_set\_nbma\_nbr\_status (struct smiclient\_globals \* azg, int ospfProcessId, struct pal\_in4\_addr neighborIpAddr, int ifIndex, int nbmaNeighborStatus, u\_int32\_t vrId)**

Sets an object that permits management of the table by facilitating actions such as row creation, construction, and destruction. smi\_ospf\_set\_nbma\_nbr\_status

**Parameters:**

← *azg* Pointer to the SMI client global structure

- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *neighborIpAddr* Neighbor address.
- ← *ifIndex* Interface index or zero.
- ← *nbmaNeighborStatus* Table management actino status
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_INCONSISTENT\_VALUE

OSPF\_API\_SET\_ERR\_WRONG\_VALUE

**2.1.2.413** `int smi_ospf_set_nbr_priority (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr neighborIpAddr, int ifIndex, int neighborPriority, u_int32_t vrId)`

Sets the priority of this neighbor in the designated router election algorithm. The value 0 signifies that the neighbor is not eligible to become the designated router on this particular network. smi\_ospf\_set\_nbr\_priority

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *neighborIpAddr* Neighbor address.
- ← *ifIndex* Interface index or zero.
- ← *neighborPriority* Priority of neighbour router.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_INCONSISTENT\_VALUE

OSPF\_API\_SET\_ERR\_WRONG\_VALUE

**2.1.2.414** `int smi_ospf_set_nssa_stability_interval (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, int nssaStabilityInterval, u_int32_t vrId)`

This function sets the number of seconds an elected translator should continue to perform its translation duties after it has determined its services are no longer required. smi\_ospf\_set\_nssa\_stability\_interval

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* OSPF area ID
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_WRONG\_VALUE  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_IS\_BACKBONE  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AREA\_NOT\_NSSA

**2.1.2.415** `int smi_ospf_set_settrap (struct smiclient_globals * azg, int ospfProcessId, int trapFlagBit, u_int32_t vrId)`

Sets the trap bit for a particular trap in ospf. smi\_ospf\_set\_settrap

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>
- int]* trapFlagBit numeric value for corresponding trap to set.
- ← *vrId* Virtual Router ID

**Returns:**

0 on success, otherwise one of the following error codes

**2.1.2.416** `int smi_ospf_set_virt_if_retrans_interval (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtIfRetransIntvl, u_int32_t vrId)`

Sets the retransmission interval time, in seconds, between link state advertisement re-transmissions, for adjacencies belonging to this interface. This value is also used when retransmitting database description and Link State request packets. This value should be well over the expected round-trip time. smi\_ospf\_set\_virt\_if\_retrans\_interval

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.



- ← *areaId* Transit Area ID.
- ← *peerId* Virtual Neighbor Router ID.
- ← *virtIfRetransIntvl* Retransmission interval in seconds.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERROR

**2.1.2.417** `int smi_ospf_set_virt_if_status (struct smiclient_globals * azg, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int virtIfStatus, u_int32_t vrId)`

Sets an object that permits management of the table facilitating actions such as row creation, construction, and destruction. smi\_ospf\_set\_virt\_if\_status

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *ospfProcessId* OSPF process ID <0-65535>.
- ← *areaId* Transit Area ID.
- ← *peerId* Virtual Neighbor Router ID.
- ← *virtIfStatus* Table management action status.
- ← *vrId* Virtual Router Id

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERROR

**2.1.2.418** `int smi_ospf_summary_address_not_advertise_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr summaryAddr, u_char summaryMask)`

This function sets the flag of the external summary address range to Not Advertise. smi\_ospf\_summary\_address\_not\_advertise\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

← *summaryAddr* The summary address

← *summaryMask* The prefix length

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_SUMMARY\_ADDRESS\_EXIST

**2.1.2.419** `int smi_ospf_summary_address_not_advertise_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr summaryAddr, u_char summaryMask)`

This function sets the flag of the external summary address range to Not Advertise. smi\_ospf\_summary\_address\_not\_advertise\_unset

#### Parameters:

← *azg* Pointer to the SMI client global structure

← *vrId* Virtual Router Id

← *ospfProcessId* OSPF process ID <0-65535>

← *ipAddr* The network address

← *summaryMask* The prefix length

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_SUMMARY\_ADDRESS\_EXIST

**2.1.2.420** `int smi_ospf_summary_address_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr summaryAddr, u_char summaryMask)`

This function sets the external summary address range. smi\_ospf\_summary\_address\_set

#### Parameters:

← *azg* Pointer to the SMI client global structure

← *vrId* Virtual Router Id

← *ospfProcessId* OSPF process ID <0-65535>

← *summaryAddr* The summary address

← *summaryMask* The prefix length

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_MALLOC\_ERR

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_SUMMARY\_ADDRESS\_EXIST

**2.1.2.421** `int smi_ospf_summary_address_tag_set (struct smiclient_globals  
* azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr  
summaryAddr, u_char summaryMask, u_int32_t tag)`

This function sets the tag value to the specified value. A tag value that can be used as a match value for controlling redistribution via route maps. smi\_ospf\_summary\_address\_tag\_set

#### Parameters:

← *azg* Pointer to the SMI client global structure

← *vrId* Virtual Router Id

← *ospfProcessId* OSPF process ID <0-65535>

← *summaryAddr* The summary address

← *summaryMask* The prefix length

← *tag* The tag value

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_SUMMARY\_ADDRESS\_EXIST

**2.1.2.422** `int smi_ospf_summary_address_tag_unset (struct smiclient_globals  
* azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr  
summaryAddr, u_char summaryMask)`

This function resets the tag value of the external summary address range to zero. smi\_ospf\_summary\_address\_tag\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *summaryAddr* The summary address
- ← *summaryMask* The prefix length

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_SUMMARY\_ADDRESS\_EXIST

**2.1.2.423** `int smi_ospf_summary_address_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr summaryAddr, u_char summaryMask)`

This function resets the external summary address range. `smi_ospf_summary_address_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *summaryAddr* The summary address
- ← *summaryMask* The prefix length

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_MALLOC\_ERR

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_SUMMARY\_ADDRESS\_EXIST

**2.1.2.424** `int smi_ospf_te_link_detail_set (struct smiclient_globals * azg, u_int32_t vrId, char * TELinkName, char * ifName, int TELinkType)`

Set detail of TE\_LINK. `smi_ospf_te_link_detail_set`

**Parameters:**

- ← *vrId* Virtual Router Id
- ← *name* TE\_LINK name
- ← *str* IPV4 address of link
- ← <0,1> 0-no Numbered 1-Numbered
- ← *azg* Pointer to the SMI client global structure

**Returns:**

SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_INVALID\_IPV4\_ADDRESS  
 OSPF\_ERR\_LINK\_NOT\_CONFIGURED

**2.1.2.425** `int smi_ospf_te_link_enable_sdkapi (struct smiclient_globals * azg, u_int32_t vr_id, char * teLinkName)`

Sets TE link mode to an OSPF process. smi\_ospf\_te\_link\_enable\_sdkapi

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *teLinkName* TE link name to be set

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERROR

**2.1.2.426** `int smi_ospf_te_link_flood_scope_set (struct smiclient_globals * azg, u_int32_t vrId, char * TELinkName, int ospfProcessId, struct pal_in4_addr areaId, int areaFormat)`

This function sets the flooding scope (area and OSPF instance) of a specified GMPLS TE link. The TE link is flooded over control links in the specified area and in the OSPF instance. smi\_ospf\_te\_link\_flood\_scope\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *TELinkName* The TE link name
- ← *ospfProcessId* OSPF Process ID for the OSPF instance <0-65535>
- ← *areaId* The area into which TE links should be flooded

← *areaFormat* Area ID format (either IP address or decimal format)

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_ABR\_TYPE\_INVALID  
 OSPF\_API\_SET\_ERR\_TELINK\_FLOOD\_SCOPE\_ENABLED

**2.1.2.427** `int smi_ospf_te_link_flood_scope_unset (struct smiclient_globals * azg, u_int32_t vrId, char * TELinkName, int ospfProcessId, struct pal_in4_addr areaId)`

This function unsets the flooding scope of a specified GMPLS TE link. If the TE link is advertised in an area, this function withdraws TE link from that area. smi\_ospf\_te\_link\_flood\_scope\_unset

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *TELinkName* The TE link name  
 ← *ospfProcessId* OSPF Process ID  
 ← *areaId* The area into which TE links should be flooded

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_ABR\_TYPE\_INVALID  
 OSPF\_API\_SET\_ERR\_TELINK\_FLOOD\_SCOPE\_NOT\_ENABLED

**2.1.2.428** `int smi_ospf_te_link_set (struct smiclient_globals * azg, u_int32_t vrId, char * TELinkName)`

Set detail of TE\_LINK. smi\_ospf\_te\_link\_set

**Parameters:**

← *vrId* Virtual Router Id  
 ← *name* TE\_LINK name

← *azg* Pointer to the SMI client global structure

**Returns:**

SUCCESS on success, otherwise one of the following error codes  
 OSPF\_ERR\_LINK\_NOT\_CONFIGURED

**2.1.2.429** `int smi_ospf_telink_te_metric_set (struct smiclient_globals * azg,  
 u_int32_t vrId, char * TELinkName, u_int32_t teMetric)`

This function sets the traffic engineering metric for a GMPLS TE-link. `smi_ospf_telink_te_metric_set`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *TELinkName* The TE link name  
 ← *teMetric* The metric value to set

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_ABR\_TYPE\_INVALID  
 OSPF\_API\_SET\_ERR\_TELINK\_METRIC\_EXIST

**2.1.2.430** `int smi_ospf_telink_te_metric_unset (struct smiclient_globals * azg,  
 u_int32_t vrId, char * TELinkName)`

This function sets the traffic engineering metric for a GMPLS TE link to the default value. `smi_ospf_telink_te_metric_unset`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *TELinkName* The TE link name

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

```

OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_ERR_ABR_TYPE_INVALID
OSPF_API_SET_ERR_TELINK_METRIC_NOT_EXIST

```

### 2.1.2.431 `int smi_ospf_timers_refresh_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, int refreshInterval)`

Sets This function sets the LSA refresh timer value. `smi_ospf_timers_refresh_set`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *refreshInterval* The refresh timer interval in seconds.<10-1800>.

#### Returns:

```

OSPF_API_SET_SUCCESSOn success, otherwise one of the following error
codes
OSPF_API_SET_ERR_PROCESS_ID_INVALID
OSPF_API_SET_ERR_PROCESS_NOT_EXIST
OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_ERR_TIMER_VALUE_INVALID

```

### 2.1.2.432 `int smi_ospf_timers_refresh_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

Sets This function resets the LSA refresh timer to the default value. `smi_ospf_timers_refresh_unset`

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

#### Returns:

```

OSPF_API_SET_SUCCESSOn success, otherwise one of the following error
codes
OSPF_API_SET_ERR_PROCESS_ID_INVALID
OSPF_API_SET_ERR_PROCESS_NOT_EXIST
OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_ERR_TIMER_VALUE_INVALID

```



**2.1.2.433** `int smi_ospf_timers_spf_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, u_int32_t spfMinDelay, u_int32_t spfMaxDelay)`

Sets This function sets the minimum and maximum delay between a topology change, being either received in an LSA or self detected, and the SPF calculation being run. `smi_ospf_timers_spf_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *spfMinDelay* The minimum SPF hold delay time in milliseconds: 500 milliseconds.
- ← *spfMaxDelay* The maximum SPF hold delay time in milliseconds: 50000 milliseconds (50 seconds).

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_TIMER\_VALUE\_INVALID

**2.1.2.434** `int smi_ospf_timers_spf_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId)`

Sets This function resets the SPF minimum delay and maximum delay to their default values: 5 seconds. `smi_ospf_timers_spf_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

**2.1.2.435** `int smi_ospf_timers_spf_validate_and_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, u_int32_t startDelay, u_int32_t minDelay, u_int32_t maxDelay)`

Sets This function sets the minimum and maximum delay between a topology change, being either received in an LSA or self detected, and the SPF calculation being run. `smi_ospf_timers_spf_validate_and_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *start\_delay* Initial SPF delay time in milliseconds:
- ← *min\_delay* The minimum SPF hold delay time in milliseconds: 500 milliseconds.
- ← *max\_delay* The maximum SPF hold delay time in milliseconds: 50000 milliseconds (50 seconds).

**Returns:**

OSPF\_API\_SET\_SUCCESS On success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_SPF\_TIMER\_MISMATCH

**2.1.2.436** `int smi_ospf_vlink_authentication_key_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, char * ifVirtAuthKey)`

This function sets the simple authentication password (type 1) for the OSPF virtual links. Simple password authentication allows a password (key) to be configured per area. To participate in the routing domain, routers in the same area must be configured with the same key. `smi_ospf_vlink_authentication_key_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID
- ← *ifVirtAuthKey* password to be used by neighbors, maximum eight characters

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST
- OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.437** `int smi_ospf_vlink_authentication_key_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId)`

This function resets the simple authentication password for the virtual link to NULL.  
smi\_ospf\_vlink\_authentication\_key\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST
- OSPF\_API\_SET\_MALLOC\_ERR
- OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.438** `int smi_ospf_vlink_authentication_type_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int ifVirtAuthType)`

This function sets the authentication type for the virtual interface. smi\_ospf\_vlink\_authentication\_type\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id

- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID
- ← *ifVirtAuthType* Authentication type ( Null | Simple password | Cryptographic)

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_AUTH\_TYPE\_INVALID  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.439** `int smi_ospf_vlink_authentication_type_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId)`

This function resets the authentication type for the virtual interface to NULL. `smi_ospf_vlink_authentication_type_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.440** `int smi_ospf_vlink_dead_interval_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int ifVirtDeadInterval)`

This function sets the router dead interval value for the virtual interface. `smi_ospf_vlink_dead_interval_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID
- ← *ifVirtDeadInterval* The dead interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

```

:* OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_ERR_PROCESS_ID_INVALID
OSPF_API_SET_ERR_PROCESS_NOT_EXIST
OSPF_API_SET_ERR_VLINK_NOT_EXIST
OSPF_API_SET_ERR_IF_DEAD_INTERVAL_INVALID
OSPF_API_SET_ERR_MALLOC_ERR

```

**2.1.2.441** `int smi_ospf_vlink_dead_interval_unset (struct smiclient_globals *  
azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId,  
struct pal_in4_addr peerId)`

This function resets the router dead interval value for the virtual interface to the default value 40. `smi_ospf_vlink_dead_interval_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

```

OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_ERR_PROCESS_ID_INVALID
OSPF_API_SET_ERR_PROCESS_NOT_EXIST
OSPF_API_SET_ERR_VLINK_NOT_EXIST
OSPF_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

```

**2.1.2.442** `int smi_ospf_vlink_hello_interval_set (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct  
pal_in4_addr peerId, int ifVirtHelloInterval)`

This function sets the router hello interval value for the virtual interface. The hello interval value must be the same for both ends of the virtual link. `smi_ospf_vlink_hello_interval_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID
- ← *ifVirtHelloInterval* The hello interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_IF\_HELLO\_INTERVAL\_INVALID
- OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.443** `int smi_ospf_vlink_hello_interval_unset (struct smiclient_globals *  
azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId,  
struct pal_in4_addr peerId)`

This function resets the router hello interval value for the virtual interface to the default value 10 seconds. `smi_ospf_vlink_hello_interval_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

```

OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_ERR_PROCESS_ID_INVALID
OSPF_API_SET_ERR_PROCESS_NOT_EXIST
OSPF_API_SET_ERR_VLINK_NOT_EXIST
OSPF_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

```

**2.1.2.444** `int smi_ospf_vlink_message_digest_key_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, u_char virtMsgDigestKeyId, char * ifVirtAuthKey)`

This function sets the MD5 authentication key for the virtual interface. `smi_ospf_vlink_message_digest_key_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID
- ← *virtMsgDigestKeyId* The key identifier.
- ← *ifVirtAuthKey* The message digest key string

**Returns:**

```

OSPF_API_SET_SUCCESS on success, otherwise one of the following error
codes
OSPF_API_SET_ERR_VR_NOT_EXIST
OSPF_API_SET_MALLOC_ERR
OSPF_API_SET_ERR_MD5_KEY_EXIST

```

**2.1.2.445** `int smi_ospf_vlink_message_digest_key_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, u_char virtMsgDigestKeyId)`

This function sets the MD5 authentication key for the virtual interface. `smi_ospf_vlink_message_digest_key_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>

- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID
- ← *virtMsgDigestKeyId* The key identifier.

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_MALLOC\_ERR

OSPF\_API\_SET\_ERR\_MD5\_KEY\_EXIST

**2.1.2.446** `int smi_ospf_vlink_retransmit_interval_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId, int retransmitInterval)`

This function sets the retransmit interval value for the virtual interface. `smi_ospf_vlink_retransmit_interval_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *area\_id* The area to which the network belongs
- ← *peer\_id* Neighbor Router ID
- ← *interval* The retransmit interval in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID

OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST

OSPF\_API\_SET\_ERR\_IF\_HELLO\_INTERVAL\_INVALID

OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.447** `int smi_ospf_vlink_retransmit_interval_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId)`

This function resets the retransmit interval value for the virtual interface to the default value 5 seconds. `smi_ospf_vlink_retransmit_interval_unset`



**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_IF\_PARAM\_NOT\_CONFIGURED

**2.1.2.448** `int smi_ospf_vlink_set (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId)`

This function creates a virtual interface and configures a virtual neighbor. smi\_ospf\_vlink\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_AREA\_NOT\_DEFAULT
- OSPF\_API\_SET\_ERR\_AREA\_LIMIT
- OSPF\_API\_SET\_ERR\_VLINK\_CANT\_GET

**2.1.2.449** `int smi_ospf_vlink_transmit_delay_set (struct smiclient_globals * azg,  
u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct  
pal_in4_addr peerId, int ifVirtTransmitDelay)`

This function sets the specified transmit delay (in seconds) for the virtual link. `smi_ospf_vlink_transmit_delay_set`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID
- ← *ifVirtTransmitDelay* The transmit delay in seconds <1-65535>

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_IF\_TRANSMIT\_DELAY\_INVALID  
 OSPF\_API\_SET\_MALLOC\_ERR

**2.1.2.450** `int smi_ospf_vlink_transmit_delay_unset (struct smiclient_globals *  
azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId,  
struct pal_in4_addr peerId)`

This function resets the transmit delay for the interface 1 seconds. `smi_ospf_vlink_transmit_delay_unset`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual Router Id
- ← *ospfProcessId* OSPF process ID <0-65535>
- ← *areaId* The area to which the network belongs
- ← *peerId* Neighbor Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

```

OSPF_API_SET_ERR_PROCESS_ID_INVALID
OSPF_API_SET_ERR_PROCESS_NOT_EXIST
OSPF_API_SET_ERR_VLINK_NOT_EXIST
OSPF_API_SET_ERR_IF_PARAM_NOT_CONFIGURED

```

**2.1.2.451** `int smi_ospf_vlink_unset (struct smiclient_globals * azg, u_int32_t vrId, int ospfProcessId, struct pal_in4_addr areaId, struct pal_in4_addr peerId)`

This function destroys the specified virtual interface and deconfigures the specified virtual neighbor. `smi_ospf_vlink_unset`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual Router Id  
 ← *ospfProcessId* OSPF process ID <0-65535>  
 ← *areaId* The area to which the network belongs  
 ← *peerId* Neighbor Router ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST      OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST

**2.1.2.452** `int smi_show_if_info (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, int start_index, int end_index, struct list * ifInfoList, int(*) (struct list * ifInfoList) funpointer)`

Fetches the OSPF process interface related information. `smi_show_if_info`

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual router-id  
 ← *ifName* OSPF process interface name  
 ← *start\_index*  
 ← *end\_index*  
 → *ifInfoList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
 RESULT\_ERROR

**2.1.2.453** `int smi_show_ospf_borderrouter_info (struct smiclient_globals * azg, u_int32_t vrId, u_int32_t ospf_id, int start_index, int end_index, struct list * brList, int(*) (struct list * brList) funpointer)`

Fetches the OSPF Border routers related information. `smi_show_ospf_borderrouter_info`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router-id
- ← *ospf\_id* OSPF process ID <0-65535>.
- ← *start\_index*
- ← *end\_index*
- *brList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
RESULT\_ERROR

**2.1.2.454** `int smi_show_ospf_buffer_info (struct smiclient_globals * azg, u_int32_t vrId, u_int16_t ospfProcessId, int start_index, int end_index, struct list * bufList, int(*) (struct list * bufList) funpointer)`

Fetches the OSPF Buffer statistics related information. `smi_show_ospf_buffer_info`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router-id
- ← *ospf\_id* OSPF process ID <0-65535>.
- ← *start\_index*
- ← *end\_index*
- *bufList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
RESULT\_ERROR

**2.1.2.455** `int smi_show_ospf_database_detail_info (struct smiclient_globals * azg, u_int32_t vrId, u_int16_t ospf_id, int start_index, int end_index, struct list * dbDetailList, int(*) (struct list * dbDetailList) funpointer)`

Fetches the OSPF process Database detailed information. `smi_show_ospf_database_detail_info`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router-id
- ← *start\_index*
- ← *end\_index*
- *dbDetailList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
 RESULT\_ERROR

**2.1.2.456** `int smi_show_ospf_database_summary_info (struct smiclient_globals * azg, u_int32_t vrId, u_int16_t ospf_id, int start_index, int end_index, struct list * dbSummaryList, int(*) (struct list *dbSummaryList) funpointer)`

Fetches the OSPF process Database information summary. `smi_show_ospf_database_summary_info`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router-id
- ← *start\_index*
- ← *end\_index*
- *dbSummaryList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
 RESULT\_ERROR

**2.1.2.457** `int smi_show_ospf_interface_brief_info (struct smiclient_globals * azg, u_int32_t vrId, char * ifName, int start_index, int end_index, struct list * ifBriefList, int(*) (struct list *ifBriefList) funpointer)`

Fetches the OSPF process interface related brief information. `smi_show_interface_brief_info`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router-id
- ← *ifName* OSPF process interface name

← *start\_index*  
 ← *end\_index*  
 → *ifInfoList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
 RESULT\_ERROR

**2.1.2.458** `int smi_show_ospf_multiarea_info (struct smiclient_globals * azg, u_int32_t vrId, u_int32_t ospf_id, int start_index, int end_index, struct list * multiAreaList, int(*) (struct list * multiAreaList) funpointer)`

Fetches the OSPF multi area adjacencies configuration information. smi\_show\_ospf\_multiarea\_info

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual router-id  
 ← *ospf\_id* OSPF process ID <0-65535>.  
 ← *start\_index*  
 ← *end\_index*  
 → *multiAreaList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
 RESULT\_ERROR

**2.1.2.459** `int smi_show_ospf_nbr_info (struct smiclient_globals * azg, u_int32_t vrId, u_int32_t ospf_id, int start_index, int end_index, struct list * nbrList, int(*) (struct list * nbrList) funpointer)`

Fetches the OSPF Neighbors related information. smi\_show\_ospf\_nbr\_info

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual router-id  
 ← *ospf\_id* OSPF process ID <0-65535>.  
 ← *start\_index*  
 ← *end\_index*  
 → *nbrList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
 RESULT\_ERROR

**2.1.2.460** `int smi_show_ospf_proc_info (struct smiclient_globals * azg,  
 u_int32_t vrId, u_int32_t ospf_id, int start_index, int end_index,  
 struct list * processList, int(*) (struct list * processList) funpointer)`

Fetches the OSPF process instance related information. smi\_show\_proc\_info

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual router-id  
 ← *ospf\_id* OSPF process ID <0-65535>.  
 ← *start\_index*  
 ← *end\_index*  
 → *ifInfoList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
 RESULT\_ERROR

**2.1.2.461** `int smi_show_ospf_route_info (struct smiclient_globals * azg,  
 u_int32_t vrId, u_int16_t ospf_id, int start_index, int end_index,  
 struct list * routeList, int(*) (struct list * routeList) funpointer)`

Fetches the OSPF Route information. smi\_show\_ospf\_route\_info

**Parameters:**

← *azg* Pointer to the SMI client global structure  
 ← *vrId* Virtual router-id  
 ← *ospf\_id* OSPF process ID <0-65535>.  
 ← *start\_index*  
 ← *end\_index*  
 → *routeList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
 RESULT\_ERROR

**2.1.2.462** `int smi_show_ospf_route_summary_info (struct smiclient_globals * azg, u_int32_t vrId, u_int16_t ospf_id, int start_index, int end_index, struct list * routeSumList, int(*) (struct list *routeSumList) funpointer)`

Fetches the OSPF Route summary information. `smi_show_ospf_route_summary_info`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router-id
- ← *ospf\_id* OSPF process ID <0-65535>.
- ← *start\_index*
- ← *end\_index*
- *routeSumList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
RESULT\_ERROR

**2.1.2.463** `int smi_show_ospf_vlink_info (struct smiclient_globals * azg, u_int32_t vrId, u_int32_t ospf_id, int start_index, int end_index, struct list * vlinkList, int(*) (struct list *vlinkList) funpointer)`

Fetches the OSPF Virtual links configuration information. `smi_show_ospf_vlink_info`

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vrId* Virtual router-id
- ← *ospf\_id* OSPF process ID <0-65535>.
- ← *start\_index*
- ← *end\_index*
- *vlinkList* Result populated list

**Returns:**

RESULT\_OK on success, otherwise one of the following error codes  
RESULT\_ERROR



## 2.2 smi\_ospf\_bfd.h File Reference

Provides APIs for managing Bidirectional Forwarding Detection(BFD) in ZebOS.

```
#include "smi_client.h"

#include "smi_ospf_bfd_msg.h"

#include "smi_ospf_msg.h"
```

### Functions

- int [smi\\_ospf\\_if\\_bfd\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*ifname)  
*Sets This function sets the BFD fall-over check for neighbors on specified interface.*
- int [smi\\_ospf\\_if\\_bfd\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*ifname)  
*Sets This function unsets the BFD fall-over check for neighbors on specified interface.*
- int [smi\\_ospf\\_if\\_bfd\\_disable\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*ifname)  
*Sets This function disables the BFD fall-over check for neighbors on specified interface.*
- int [smi\\_ospf\\_if\\_bfd\\_disable\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*ifname)  
*Sets This function unset the disable flag of BFD fall-over check for neighbors on specified interface.*
- int [smi\\_ospf\\_bfd\\_all\\_interfaces\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, int proc\_id)
- int [smi\\_ospf\\_bfd\\_all\\_interfaces\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, int proc\_id)  
*Sets This function unset the BFD fall-over check for all the neighbors under specified process.*
- int [smi\\_ospf\\_vlink\\_bfd\\_set](#) (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, int proc\_id, struct pal\_in4\_addr area\_id, struct pal\_in4\_addr peer\_id)  
*Sets This function set the BFD fall-over check for the specified VLINK neighbor.*
- int [smi\\_ospf\\_vlink\\_bfd\\_unset](#) (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, int proc\_id, struct pal\_in4\_addr area\_id, struct pal\_in4\_addr peer\_id)  
*Sets This function unsets the BFD fall-over check for the specified VLINK neighbor.*
- int [smi\\_ospf\\_if\\_bfd\\_set\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- int [smi\\_ospf\\_if\\_bfd\\_disable\\_set\\_validate](#) (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)

- **int smi\_ospf\_bfd\_all\_interfaces\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_vlink\_bfd\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)
- **int smi\_ospf\_if\_bfd\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- **int smi\_ospf\_if\_bfd\_disable\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, char \*ifName)
- **int smi\_ospf\_bfd\_all\_interfaces\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId)
- **int smi\_ospf\_vlink\_bfd\_unset\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vrId, int ospfProcessId, struct pal\_in4\_addr areaId, struct pal\_in4\_addr peerId)

### 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\_ospf\_bfd\_all\_interfaces\_set (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, int proc\_id)

smi\_ospf\_bfd\_all\_interfaces\_set

@ brief Sets This function sets the BFD fall-over check for all the neighbors under specified process.

##### Parameters:

- ← **azg** Pointer to the SMI client global structure
- ← **vr\_id** Virtual Router Id
- ← **proc\_id** OSPF process ID

##### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
 OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
 OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID  
 OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

#### 2.2.2.2 int smi\_ospf\_bfd\_all\_interfaces\_unset (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, int proc\_id)

Sets This function unset the BFD fall-over check for all the neighbors under specified process. smi\_ospf\_bfd\_all\_interfaces\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router Id
- ← *proc\_id* OSPF process ID

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST

**2.2.2.3 int smi\_ospf\_if\_bfd\_disable\_set (struct smiclient\_globals \* *azg*,  
u\_int32\_t *vr\_id*, char \* *ifname*)**

Sets This function disables the BFD fall-over check for neighbors on specified interface. smi\_ospf\_if\_bfd\_disable\_set

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router Id
- ← *ifname* The interface name

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_MALLOC\_ERR

**2.2.2.4 int smi\_ospf\_if\_bfd\_disable\_unset (struct smiclient\_globals \* *azg*,  
u\_int32\_t *vr\_id*, char \* *ifname*)**

Sets This function unset the disable flag of BFD fall-over check for neighbors on specified interface. smi\_ospf\_if\_bfd\_disable\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router Id
- ← *ifname* The interface name

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.2.2.5 `int smi_ospf_if_bfd_set (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname)`

Sets This function sets the BFD fall-over check for neighbors on specified interface.  
smi\_ospf\_if\_bfd\_set

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router Id
- ← *ifname* The interface name

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST  
OSPF\_API\_SET\_MALLOC\_ERR

### 2.2.2.6 `int smi_ospf_if_bfd_unset (struct smiclient_globals * azg, u_int32_t vr_id, char * ifname)`

Sets This function unsets the BFD fall-over check for neighbors on specified interface.  
smi\_ospf\_if\_bfd\_unset

#### Parameters:

- ← *azg* Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router Id
- ← *ifname* The interface name

#### Returns:

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes  
OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST

### 2.2.2.7 `int smi_ospf_vlink_bfd_set (struct smiclient_globals * azg, u_int32_t vr_id, int proc_id, struct pal_in4_addr area_id, struct pal_in4_addr peer_id)`

Sets This function set the BFD fall-over check for the specified VLINK neighbor. smi\_ospf\_vlink\_bfd\_set

#### Parameters:

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

- ← *vr\_id* Virtual Router Id
- ← *proc\_id* OSPF process ID
- ← *area\_id* Area id
- ← *peer\_id* Peer id

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST

**2.2.2.8** `int smi_ospf_vlink_bfd_unset (struct smiclient_globals * azg, u_int32_t vr_id, int proc_id, struct pal_in4_addr area_id, struct pal_in4_addr peer_id)`

Sets This function unsets the BFD fall-over check for the specified VLINK neighbor.  
smi\_ospf\_vlink\_bfd\_unset

**Parameters:**

- ← *azg* Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router Id
- ← *proc\_id* OSPF process ID
- ← *area\_id* Area id
- ← *peer\_id* Peer id

**Returns:**

OSPF\_API\_SET\_SUCCESS on success, otherwise one of the following error codes

- OSPF\_API\_SET\_ERR\_VR\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_PROCESS\_ID\_INVALID
- OSPF\_API\_SET\_ERR\_PROCESS\_NOT\_EXIST
- OSPF\_API\_SET\_ERR\_VLINK\_NOT\_EXIST

# Index

smi\_debug\_ospf\_packet\_set  
    smi\_ospf.h, [71](#)  
smi\_debug\_ospf\_packet\_unset  
    smi\_ospf.h, [71](#)  
smi\_ospf.h, [3](#)  
    smi\_debug\_ospf\_packet\_set, [71](#)  
    smi\_debug\_ospf\_packet\_unset, [71](#)  
    smi\_ospf\_abr\_type\_set, [72](#)  
    smi\_ospf\_abr\_type\_unset, [72](#)  
    smi\_ospf\_area\_auth\_by\_type\_unset,  
        [73](#)  
    smi\_ospf\_area\_auth\_type\_set, [73](#)  
    smi\_ospf\_area\_auth\_type\_unset, [74](#)  
    smi\_ospf\_area\_default\_cost\_set, [74](#)  
    smi\_ospf\_area\_default\_cost\_unset,  
        [75](#)  
    smi\_ospf\_area\_default\_cost\_value\_-  
        unset, [75](#)  
    smi\_ospf\_area\_export\_list\_set, [76](#)  
    smi\_ospf\_area\_export\_list\_unset, [76](#)  
    smi\_ospf\_area\_filter\_list\_access\_-  
        set, [77](#)  
    smi\_ospf\_area\_filter\_list\_access\_-  
        unset, [77](#)  
    smi\_ospf\_area\_filter\_list\_prefix\_set,  
        [78](#)  
    smi\_ospf\_area\_filter\_list\_prefix\_-  
        unset, [78](#)  
    smi\_ospf\_area\_import\_list\_set, [79](#)  
    smi\_ospf\_area\_import\_list\_unset,  
        [79](#)  
    smi\_ospf\_area\_no\_summary\_set, [80](#)  
    smi\_ospf\_area\_no\_summary\_unset,  
        [80](#)  
    smi\_ospf\_area\_nssa\_default\_-  
        originate\_metric\_set, [81](#)  
    smi\_ospf\_area\_nssa\_default\_-  
        originate\_metric\_type\_set,  
        [81](#)  
    smi\_ospf\_area\_nssa\_default\_-  
        originate\_route\_map\_set, [82](#)  
    smi\_ospf\_area\_nssa\_default\_-  
        originate\_set, [83](#)  
    smi\_ospf\_area\_nssa\_default\_-  
        originate\_unset, [83](#)  
    smi\_ospf\_area\_nssa\_no\_-  
        redistribution\_set, [84](#)  
    smi\_ospf\_area\_nssa\_no\_-  
        redistribution\_unset, [84](#)  
    smi\_ospf\_area\_nssa\_set, [85](#)  
    smi\_ospf\_area\_nssa\_stability\_-  
        interval\_set, [85](#)  
    smi\_ospf\_area\_nssa\_translator\_-  
        role\_set, [86](#)  
    smi\_ospf\_area\_nssa\_translator\_-  
        role\_unset, [86](#)  
    smi\_ospf\_area\_nssa\_unset, [87](#)  
    smi\_ospf\_area\_range\_not\_-  
        advertise\_set, [87](#)  
    smi\_ospf\_area\_range\_not\_-  
        advertise\_unset, [88](#)  
    smi\_ospf\_area\_range\_set, [88](#)  
    smi\_ospf\_area\_range\_substitute\_set,  
        [89](#)  
    smi\_ospf\_area\_range\_substitute\_-  
        unset, [90](#)  
    smi\_ospf\_area\_range\_unset, [90](#)  
    smi\_ospf\_area\_shortcut\_set, [91](#)  
    smi\_ospf\_area\_shortcut\_unset, [91](#)  
    smi\_ospf\_area\_stub\_set, [92](#)  
    smi\_ospf\_area\_stub\_unset, [92](#)  
    smi\_ospf\_auto\_cost\_reference\_-  
        bandwidth\_set, [93](#)  
    smi\_ospf\_auto\_cost\_reference\_-  
        bandwidth\_type\_set, [93](#)  
    smi\_ospf\_auto\_cost\_reference\_-  
        bandwidth\_unset, [94](#)  
    smi\_ospf\_capability\_cspf\_set, [94](#)  
    smi\_ospf\_capability\_cspf\_unset, [95](#)  
    smi\_ospf\_capability\_opaque\_lsa\_-  
        set, [95](#)

- smi\_ospf\_capability\_opaque\_lsa\_-unset, [96](#)
- smi\_ospf\_capability\_restart\_set, [96](#)
- smi\_ospf\_capability\_restart\_unset, [96](#)
- smi\_ospf\_capability\_traffic\_-engineering\_set, [97](#)
- smi\_ospf\_capability\_traffic\_-engineering\_unset, [97](#)
- smi\_ospf\_compatible\_rfc1583\_set, [98](#)
- smi\_ospf\_compatible\_rfc1583\_-unset, [98](#)
- smi\_ospf\_cspf\_better\_protection\_-type, [99](#)
- smi\_ospf\_debug\_set, [99](#)
- smi\_ospf\_debug\_unset, [101](#)
- smi\_ospf\_default\_metric\_set, [102](#)
- smi\_ospf\_default\_metric\_unset, [103](#)
- smi\_ospf\_disable\_db\_summary\_opt, [103](#)
- smi\_ospf\_disable\_ext\_multi\_inst, [103](#)
- smi\_ospf\_distance\_all\_set, [104](#)
- smi\_ospf\_distance\_all\_unset, [104](#)
- smi\_ospf\_distance\_external\_set, [105](#)
- smi\_ospf\_distance\_external\_unset, [105](#)
- smi\_ospf\_distance\_inter\_area\_set, [106](#)
- smi\_ospf\_distance\_inter\_area\_-unset, [106](#)
- smi\_ospf\_distance\_intra\_area\_set, [107](#)
- smi\_ospf\_distance\_intra\_area\_-unset, [107](#)
- smi\_ospf\_distance\_source\_set, [107](#)
- smi\_ospf\_distance\_source\_unset, [108](#)
- smi\_ospf\_distribute\_list\_in\_set, [109](#)
- smi\_ospf\_distribute\_list\_in\_unset, [109](#)
- smi\_ospf\_distribute\_list\_out\_set, [109](#)
- smi\_ospf\_distribute\_list\_out\_unset, [110](#)
- smi\_ospf\_dna\_set\_sdkapi, [111](#)
- smi\_ospf\_dna\_unset\_sdkapi, [111](#)
- smi\_ospf\_domain\_id\_set, [111](#)
- smi\_ospf\_domain\_id\_unset, [112](#)
- smi\_ospf\_enable\_db\_summary\_opt, [112](#)
- smi\_ospf\_enable\_ext\_multi\_inst, [113](#)
- smi\_ospf\_get\_address\_less\_if, [113](#)
- smi\_ospf\_get\_admin\_stat, [114](#)
- smi\_ospf\_get\_area\_aggregate\_-area\_id, [114](#)
- smi\_ospf\_get\_area\_aggregate\_-effect, [115](#)
- smi\_ospf\_get\_area\_aggregate\_-lsdb\_type, [115](#)
- smi\_ospf\_get\_area\_aggregate\_-mask, [116](#)
- smi\_ospf\_get\_area\_aggregate\_net, [116](#)
- smi\_ospf\_get\_area\_aggregate\_-route\_tag, [117](#)
- smi\_ospf\_get\_area\_aggregate\_-status, [117](#)
- smi\_ospf\_get\_area\_bdr\_rtr\_count, [118](#)
- smi\_ospf\_get\_area\_bdr\_rtr\_status, [118](#)
- smi\_ospf\_get\_area\_id, [119](#)
- smi\_ospf\_get\_area\_lsa\_cksum\_sum, [119](#)
- smi\_ospf\_get\_area\_lsa\_count, [120](#)
- smi\_ospf\_get\_area\_lsa\_count\_-number, [120](#)
- smi\_ospf\_get\_area\_nssa\_-translator\_events, [120](#)
- smi\_ospf\_get\_area\_nssa\_-translator\_role, [121](#)
- smi\_ospf\_get\_area\_nssa\_-translator\_stability\_interval, [121](#)
- smi\_ospf\_get\_area\_nssa\_-translator\_state, [122](#)
- smi\_ospf\_get\_area\_range\_area\_id, [122](#)
- smi\_ospf\_get\_area\_range\_effect, [123](#)
- smi\_ospf\_get\_area\_range\_mask, [123](#)
- smi\_ospf\_get\_area\_range\_net, [124](#)
- smi\_ospf\_get\_area\_range\_status, [124](#)
- smi\_ospf\_get\_area\_status, [125](#)
- smi\_ospf\_get\_area\_summary, [125](#)

- smi\_ospf\_get\_as\_lsdb\_age, [125](#)
- smi\_ospf\_get\_as\_lsdb\_checksum, [126](#)
- smi\_ospf\_get\_as\_lsdb\_sequence, [126](#)
- smi\_ospf\_get\_as\_scope\_lsa\_count, [127](#)
- smi\_ospf\_get\_asbdr\_rtr\_count, [127](#)
- smi\_ospf\_get\_asbdr\_rtr\_status, [128](#)
- smi\_ospf\_get\_auth\_type, [128](#)
- smi\_ospf\_get\_compatible\_rfc1583, [129](#)
- smi\_ospf\_get\_demand\_extensions, [129](#)
- smi\_ospf\_get\_discontinuity\_time, [129](#)
- smi\_ospf\_get\_domain\_id, [130](#)
- smi\_ospf\_get\_exit\_overflow\_interval, [130](#)
- smi\_ospf\_get\_ext\_lsdb\_advertisement, [131](#)
- smi\_ospf\_get\_ext\_lsdb\_age, [131](#)
- smi\_ospf\_get\_ext\_lsdb\_checksum, [132](#)
- smi\_ospf\_get\_ext\_lsdb\_limit, [132](#)
- smi\_ospf\_get\_ext\_lsdb\_lsid, [133](#)
- smi\_ospf\_get\_ext\_lsdb\_router\_id, [133](#)
- smi\_ospf\_get\_ext\_lsdb\_sequence, [134](#)
- smi\_ospf\_get\_ext\_lsdb\_type, [134](#)
- smi\_ospf\_get\_extn\_lsa\_cksum\_sum, [135](#)
- smi\_ospf\_get\_extn\_lsa\_count, [135](#)
- smi\_ospf\_get\_external\_type1\_metric, [136](#)
- smi\_ospf\_get\_external\_type2\_metric, [136](#)
- smi\_ospf\_get\_host\_area\_id, [136](#)
- smi\_ospf\_get\_host\_cfg\_area\_id, [137](#)
- smi\_ospf\_get\_host\_ip\_address, [137](#)
- smi\_ospf\_get\_host\_metric, [138](#)
- smi\_ospf\_get\_host\_status, [138](#)
- smi\_ospf\_get\_host\_tos, [139](#)
- smi\_ospf\_get\_if\_admin\_stat, [139](#)
- smi\_ospf\_get\_if\_area\_id, [140](#)
- smi\_ospf\_get\_if\_auth\_key, [140](#)
- smi\_ospf\_get\_if\_auth\_type, [141](#)
- smi\_ospf\_get\_if\_backup\_designated\_router, [141](#)
- smi\_ospf\_get\_if\_bdr, [142](#)
- smi\_ospf\_get\_if\_demand, [142](#)
- smi\_ospf\_get\_if\_designated\_router, [143](#)
- smi\_ospf\_get\_if\_dr, [143](#)
- smi\_ospf\_get\_if\_events, [143](#)
- smi\_ospf\_get\_if\_hello\_interval, [144](#)
- smi\_ospf\_get\_if\_ip\_address, [144](#)
- smi\_ospf\_get\_if\_lsa\_checksum, [145](#)
- smi\_ospf\_get\_if\_lsa\_count, [145](#)
- smi\_ospf\_get\_if\_metric\_address\_less\_if, [146](#)
- smi\_ospf\_get\_if\_metric\_ip\_address, [146](#)
- smi\_ospf\_get\_if\_metric\_status, [147](#)
- smi\_ospf\_get\_if\_metric\_value, [147](#)
- smi\_ospf\_get\_if\_multicast\_forwarding, [148](#)
- smi\_ospf\_get\_if\_poll\_interval, [148](#)
- smi\_ospf\_get\_if\_retrans\_interval, [149](#)
- smi\_ospf\_get\_if\_rtr\_dead\_interval, [149](#)
- smi\_ospf\_get\_if\_rtr\_priority, [150](#)
- smi\_ospf\_get\_if\_state, [150](#)
- smi\_ospf\_get\_if\_status, [151](#)
- smi\_ospf\_get\_if\_transit\_delay, [151](#)
- smi\_ospf\_get\_if\_type, [152](#)
- smi\_ospf\_get\_import\_as\_extern, [153](#)
- smi\_ospf\_get\_inter\_area\_metric, [153](#)
- smi\_ospf\_get\_intra\_area\_metric, [153](#)
- smi\_ospf\_get\_local\_lsdb\_age, [154](#)
- smi\_ospf\_get\_local\_lsdb\_checksum, [154](#)
- smi\_ospf\_get\_local\_lsdb\_sequence, [155](#)
- smi\_ospf\_get\_lsdb\_advertisement, [155](#)
- smi\_ospf\_get\_lsdb\_age, [156](#)
- smi\_ospf\_get\_lsdb\_area\_id, [156](#)
- smi\_ospf\_get\_lsdb\_checksum, [157](#)
- smi\_ospf\_get\_lsdb\_lsid, [158](#)
- smi\_ospf\_get\_lsdb\_router\_id, [158](#)
- smi\_ospf\_get\_lsdb\_sequence, [159](#)
- smi\_ospf\_get\_lsdb\_type, [159](#)
- smi\_ospf\_get\_nbma\_nbr\_permanence, [160](#)



- smi\_ospf\_get\_nbma\_nbr\_status, 160
- smi\_ospf\_get\_nbr\_address\_less\_-  
index, 161
- smi\_ospf\_get\_nbr\_events, 161
- smi\_ospf\_get\_nbr\_hello\_-  
suppressed, 162
- smi\_ospf\_get\_nbr\_ip\_addr, 162
- smi\_ospf\_get\_nbr\_ls\_retrans\_qlen,  
163
- smi\_ospf\_get\_nbr\_options, 163
- smi\_ospf\_get\_nbr\_priority, 164
- smi\_ospf\_get\_nbr\_restart\_helper\_-  
age, 164
- smi\_ospf\_get\_nbr\_restart\_helper\_-  
exit\_reason, 165
- smi\_ospf\_get\_nbr\_restart\_helper\_-  
status, 165
- smi\_ospf\_get\_nbr\_rtr\_id, 166
- smi\_ospf\_get\_nbr\_state, 166
- smi\_ospf\_get\_opaque\_lsa\_support,  
167
- smi\_ospf\_get\_originate\_new\_lsas,  
167
- smi\_ospf\_get\_reference\_bandwidth,  
168
- smi\_ospf\_get\_restart\_age, 168
- smi\_ospf\_get\_restart\_exit\_reason,  
169
- smi\_ospf\_get\_restart\_interval, 169
- smi\_ospf\_get\_restart\_status, 169
- smi\_ospf\_get\_restart\_strict\_lsa\_-  
check, 170
- smi\_ospf\_get\_restart\_support, 170
- smi\_ospf\_get\_router\_id, 171
- smi\_ospf\_get\_rx\_new\_lsas, 171
- smi\_ospf\_get\_settrap, 171
- smi\_ospf\_get\_spf\_runs, 172
- smi\_ospf\_get\_stub\_area\_id, 172
- smi\_ospf\_get\_stub\_metric, 173
- smi\_ospf\_get\_stub\_metric\_type,  
173
- smi\_ospf\_get\_stub\_router\_-  
advertisement, 174
- smi\_ospf\_get\_stub\_router\_support,  
174
- smi\_ospf\_get\_stub\_status, 174
- smi\_ospf\_get\_stub\_tos, 175
- smi\_ospf\_get\_tos\_support, 175
- smi\_ospf\_get\_version\_number, 176
- smi\_ospf\_get\_virt\_if\_area\_id, 176
- smi\_ospf\_get\_virt\_if\_auth\_key, 177
- smi\_ospf\_get\_virt\_if\_auth\_type,  
177
- smi\_ospf\_get\_virt\_if\_events, 177
- smi\_ospf\_get\_virt\_if\_hello\_interval,  
178
- smi\_ospf\_get\_virt\_if\_lsa\_-  
cksumsum, 178
- smi\_ospf\_get\_virt\_if\_lsa\_count, 179
- smi\_ospf\_get\_virt\_if\_neighbor, 179
- smi\_ospf\_get\_virt\_if\_retrans\_-  
interval, 180
- smi\_ospf\_get\_virt\_if\_rtr\_dead\_-  
interval, 180
- smi\_ospf\_get\_virt\_if\_state, 181
- smi\_ospf\_get\_virt\_if\_status, 181
- smi\_ospf\_get\_virt\_if\_transit\_delay,  
182
- smi\_ospf\_get\_virt\_local\_lsdb\_age,  
182
- smi\_ospf\_get\_virt\_local\_lsdb\_-  
checksum, 183
- smi\_ospf\_get\_virt\_local\_lsdb\_-  
sequence, 183
- smi\_ospf\_get\_virt\_nbr\_area, 184
- smi\_ospf\_get\_virt\_nbr\_events, 184
- smi\_ospf\_get\_virt\_nbr\_hello\_-  
suppressed, 185
- smi\_ospf\_get\_virt\_nbr\_ip\_addr, 185
- smi\_ospf\_get\_virt\_nbr\_ls\_retrans\_-  
qlen, 186
- smi\_ospf\_get\_virt\_nbr\_options, 186
- smi\_ospf\_get\_virt\_nbr\_restart\_-  
helper\_age, 187
- smi\_ospf\_get\_virt\_nbr\_restart\_-  
helper\_exit\_reason, 187
- smi\_ospf\_get\_virt\_nbr\_restart\_-  
helper\_status, 188
- smi\_ospf\_get\_virt\_nbr\_rtr\_id, 188
- smi\_ospf\_get\_virt\_nbr\_state, 189
- smi\_ospf\_graceful\_restart\_-  
planned\_set\_sdkapi, 189
- smi\_ospf\_graceful\_restart\_-  
planned\_unset\_sdkapi, 190
- smi\_ospf\_graceful\_restart\_set\_-  
sdkapi, 190
- smi\_ospf\_graceful\_restart\_unset\_-  
sdkapi, 190
- smi\_ospf\_host\_entry\_cost\_set, 191

- smi\_ospf\_host\_entry\_cost\_unset, 191
- smi\_ospf\_host\_entry\_set, 192
- smi\_ospf\_host\_entry\_unset, 192
- smi\_ospf\_if\_authentication\_key\_set, 193
- smi\_ospf\_if\_authentication\_key\_set\_by\_addr, 193
- smi\_ospf\_if\_authentication\_key\_unset, 194
- smi\_ospf\_if\_authentication\_key\_unset\_by\_addr, 194
- smi\_ospf\_if\_authentication\_type\_set, 195
- smi\_ospf\_if\_authentication\_type\_set\_by\_addr, 195
- smi\_ospf\_if\_authentication\_type\_unset, 196
- smi\_ospf\_if\_authentication\_type\_unset\_by\_addr, 196
- smi\_ospf\_if\_conf\_ldp\_igp\_set\_sdkapi, 196
- smi\_ospf\_if\_conf\_ldp\_igp\_unset\_sdkapi, 197
- smi\_ospf\_if\_cost\_set, 197
- smi\_ospf\_if\_cost\_set\_by\_addr, 198
- smi\_ospf\_if\_cost\_unset, 198
- smi\_ospf\_if\_cost\_unset\_by\_addr, 198
- smi\_ospf\_if\_cost\_value\_unset, 199
- smi\_ospf\_if\_database\_filter\_set, 199
- smi\_ospf\_if\_database\_filter\_set\_by\_addr, 200
- smi\_ospf\_if\_database\_filter\_unset, 200
- smi\_ospf\_if\_database\_filter\_unset\_by\_addr, 200
- smi\_ospf\_if\_dead\_interval\_set, 201
- smi\_ospf\_if\_dead\_interval\_set\_by\_addr, 201
- smi\_ospf\_if\_dead\_interval\_unset, 202
- smi\_ospf\_if\_dead\_interval\_unset\_by\_addr, 202
- smi\_ospf\_if\_disable\_all\_set, 203
- smi\_ospf\_if\_disable\_all\_unset, 203
- smi\_ospf\_if\_dna\_set, 203
- smi\_ospf\_if\_dna\_unset, 204
- smi\_ospf\_if\_hello\_interval\_set, 204
- smi\_ospf\_if\_hello\_interval\_set\_by\_addr, 205
- smi\_ospf\_if\_hello\_interval\_unset, 205
- smi\_ospf\_if\_hello\_interval\_unset\_by\_addr, 206
- smi\_ospf\_if\_ip\_router\_set, 206
- smi\_ospf\_if\_ip\_router\_unset, 207
- smi\_ospf\_if\_message\_digest\_key\_get, 207
- smi\_ospf\_if\_message\_digest\_key\_set, 208
- smi\_ospf\_if\_message\_digest\_key\_set\_by\_addr, 208
- smi\_ospf\_if\_message\_digest\_key\_unset, 209
- smi\_ospf\_if\_message\_digest\_key\_unset\_by\_addr, 209
- smi\_ospf\_if\_mtu\_ignore\_set, 210
- smi\_ospf\_if\_mtu\_ignore\_set\_by\_addr, 210
- smi\_ospf\_if\_mtu\_ignore\_unset, 211
- smi\_ospf\_if\_mtu\_ignore\_unset\_by\_addr, 211
- smi\_ospf\_if\_mtu\_set, 211
- smi\_ospf\_if\_mtu\_unset, 212
- smi\_ospf\_if\_network\_p2mp\_nbma\_set, 212
- smi\_ospf\_if\_network\_set, 213
- smi\_ospf\_if\_network\_unset, 213
- smi\_ospf\_if\_passive\_interface\_set, 213
- smi\_ospf\_if\_passive\_interface\_unset, 214
- smi\_ospf\_if\_priority\_set, 214
- smi\_ospf\_if\_priority\_set\_by\_addr, 215
- smi\_ospf\_if\_priority\_unset, 215
- smi\_ospf\_if\_priority\_unset\_by\_addr, 215
- smi\_ospf\_if\_resync\_timeout\_set, 216
- smi\_ospf\_if\_resync\_timeout\_set\_by\_addr, 216
- smi\_ospf\_if\_resync\_timeout\_unset, 217
- smi\_ospf\_if\_resync\_timeout\_unset\_by\_addr, 217
- smi\_ospf\_if\_retransmit\_interval\_set, 218

- smi\_ospf\_if\_retransmit\_interval\_-  
set\_by\_addr, [218](#)
- smi\_ospf\_if\_retransmit\_interval\_-  
unset, [218](#)
- smi\_ospf\_if\_retransmit\_interval\_-  
unset\_by\_addr, [219](#)
- smi\_ospf\_if\_te\_metric\_set, [219](#)
- smi\_ospf\_if\_te\_metric\_unset, [220](#)
- smi\_ospf\_if\_transmit\_delay\_set,  
[220](#)
- smi\_ospf\_if\_transmit\_delay\_set\_-  
by\_addr, [221](#)
- smi\_ospf\_if\_transmit\_delay\_unset,  
[221](#)
- smi\_ospf\_if\_transmit\_delay\_unset\_-  
by\_addr, [221](#)
- smi\_ospf\_log\_adj\_changes\_set, [222](#)
- smi\_ospf\_log\_adj\_changes\_unset,  
[222](#)
- smi\_ospf\_lsa\_min\_arrival\_set, [223](#)
- smi\_ospf\_lsa\_min\_arrival\_unset,  
[223](#)
- smi\_ospf\_lsa\_throttle\_timers\_set,  
[224](#)
- smi\_ospf\_lsa\_throttle\_timers\_unset,  
[224](#)
- smi\_ospf\_max\_area\_limit\_set\_-  
sdkapi, [225](#)
- smi\_ospf\_max\_area\_limit\_unset\_-  
sdkapi, [225](#)
- smi\_ospf\_max\_concurrent\_dd\_set,  
[225](#)
- smi\_ospf\_max\_concurrent\_dd\_-  
unset, [226](#)
- smi\_ospf\_max\_unuse\_lsa\_set, [226](#)
- smi\_ospf\_max\_unuse\_lsa\_unset,  
[227](#)
- smi\_ospf\_max\_unuse\_packet\_set,  
[227](#)
- smi\_ospf\_max\_unuse\_packet\_unset,  
[227](#)
- smi\_ospf\_multi\_area\_adjacency\_-  
set, [228](#)
- smi\_ospf\_multi\_area\_adjacency\_-  
unset, [228](#)
- smi\_ospf\_nbr\_static\_cost\_set, [229](#)
- smi\_ospf\_nbr\_static\_cost\_unset,  
[229](#)
- smi\_ospf\_nbr\_static\_poll\_interval\_-  
set, [230](#)
- smi\_ospf\_nbr\_static\_poll\_interval\_-  
unset, [230](#)
- smi\_ospf\_nbr\_static\_priority\_set,  
[231](#)
- smi\_ospf\_nbr\_static\_priority\_unset,  
[231](#)
- smi\_ospf\_nbr\_static\_set, [232](#)
- smi\_ospf\_nbr\_static\_unset, [232](#)
- smi\_ospf\_network\_set, [233](#)
- smi\_ospf\_network\_unset, [234](#)
- smi\_ospf\_opaque\_area\_lsa\_set, [234](#)
- smi\_ospf\_opaque\_as\_lsa\_set, [235](#)
- smi\_ospf\_opaque\_link\_lsa\_set, [235](#)
- smi\_ospf\_opaque\_te\_link\_local\_-  
lsa\_disable, [236](#)
- smi\_ospf\_opaque\_te\_link\_local\_-  
lsa\_enable, [236](#)
- smi\_ospf\_overflow\_database\_-  
external\_interval\_set, [237](#)
- smi\_ospf\_overflow\_database\_-  
external\_interval\_unset, [237](#)
- smi\_ospf\_overflow\_database\_-  
external\_limit\_set, [238](#)
- smi\_ospf\_overflow\_database\_-  
external\_limit\_unset, [238](#)
- smi\_ospf\_passive\_interface\_-  
default\_set, [239](#)
- smi\_ospf\_passive\_interface\_-  
default\_unset, [239](#)
- smi\_ospf\_passive\_interface\_set, [239](#)
- smi\_ospf\_passive\_interface\_set\_-  
by\_addr, [240](#)
- smi\_ospf\_passive\_interface\_unset,  
[240](#)
- smi\_ospf\_passive\_interface\_unset\_-  
by\_addr, [241](#)
- smi\_ospf\_process\_set, [241](#)
- smi\_ospf\_process\_set\_vrf, [242](#)
- smi\_ospf\_process\_shut\_set, [242](#)
- smi\_ospf\_process\_shut\_unset, [243](#)
- smi\_ospf\_process\_unset, [243](#)
- smi\_ospf\_redis\_default\_set, [243](#)
- smi\_ospf\_redis\_default\_unset, [244](#)
- smi\_ospf\_redis\_metric\_set, [244](#)
- smi\_ospf\_redis\_metric\_type\_set,  
[245](#)
- smi\_ospf\_redis\_metric\_type\_unset,  
[246](#)
- smi\_ospf\_redis\_metric\_unset, [246](#)
- smi\_ospf\_redis\_proto\_set, [247](#)

- smi\_ospf\_redist\_proto\_unset, [247](#)
- smi\_ospf\_redist\_tag\_set, [248](#)
- smi\_ospf\_redist\_tag\_unset, [248](#)
- smi\_ospf\_redistribute\_default\_set, [249](#)
- smi\_ospf\_redistribute\_set, [249](#)
- smi\_ospf\_restart\_graceful\_sdkapi, [250](#)
- smi\_ospf\_restart\_helper\_grace\_-period\_set, [251](#)
- smi\_ospf\_restart\_helper\_grace\_-period\_unset, [251](#)
- smi\_ospf\_restart\_helper\_never\_-router\_set, [251](#)
- smi\_ospf\_restart\_helper\_never\_-router\_unset, [252](#)
- smi\_ospf\_restart\_helper\_never\_-router\_unset\_all, [252](#)
- smi\_ospf\_restart\_helper\_policy\_set, [252](#)
- smi\_ospf\_restart\_helper\_policy\_-unset, [253](#)
- smi\_ospf\_routemap\_default\_set, [253](#)
- smi\_ospf\_routemap\_default\_unset, [254](#)
- smi\_ospf\_routemap\_set, [254](#)
- smi\_ospf\_routemap\_unset, [255](#)
- smi\_ospf\_router\_id\_set, [255](#)
- smi\_ospf\_router\_id\_unset, [256](#)
- smi\_ospf\_set\_area\_aggregate\_-effect, [256](#)
- smi\_ospf\_set\_area\_aggregate\_-route\_tag, [257](#)
- smi\_ospf\_set\_area\_aggregate\_-status, [257](#)
- smi\_ospf\_set\_area\_status, [258](#)
- smi\_ospf\_set\_asbdr\_rtr\_status, [258](#)
- smi\_ospf\_set\_if\_admin\_stat, [259](#)
- smi\_ospf\_set\_if\_area\_id, [259](#)
- smi\_ospf\_set\_if\_auth\_key, [260](#)
- smi\_ospf\_set\_if\_auth\_type, [260](#)
- smi\_ospf\_set\_if\_hello\_interval, [261](#)
- smi\_ospf\_set\_if\_metric\_status, [261](#)
- smi\_ospf\_set\_if\_metric\_value, [262](#)
- smi\_ospf\_set\_if\_poll\_interval, [262](#)
- smi\_ospf\_set\_if\_retrans\_interval, [263](#)
- smi\_ospf\_set\_if\_rtr\_dead\_interval, [264](#)
- smi\_ospf\_set\_if\_rtr\_priority, [264](#)
- smi\_ospf\_set\_if\_status, [265](#)
- smi\_ospf\_set\_if\_transit\_delay, [265](#)
- smi\_ospf\_set\_if\_type, [266](#)
- smi\_ospf\_set\_import\_as\_extern, [266](#)
- smi\_ospf\_set\_lsdb\_limit\_sdkapi, [267](#)
- smi\_ospf\_set\_multicast\_extensions, [268](#)
- smi\_ospf\_set\_nbma\_nbr\_status, [268](#)
- smi\_ospf\_set\_nbr\_priority, [269](#)
- smi\_ospf\_set\_nssa\_stability\_-interval, [269](#)
- smi\_ospf\_set\_settrap, [270](#)
- smi\_ospf\_set\_virt\_if\_retrans\_-interval, [270](#)
- smi\_ospf\_set\_virt\_if\_status, [271](#)
- smi\_ospf\_summary\_address\_not\_-advertise\_set, [271](#)
- smi\_ospf\_summary\_address\_not\_-advertise\_unset, [272](#)
- smi\_ospf\_summary\_address\_set, [272](#)
- smi\_ospf\_summary\_address\_tag\_-set, [273](#)
- smi\_ospf\_summary\_address\_tag\_-unset, [273](#)
- smi\_ospf\_summary\_address\_unset, [274](#)
- smi\_ospf\_te\_link\_detail\_set, [274](#)
- smi\_ospf\_te\_link\_enable\_sdkapi, [275](#)
- smi\_ospf\_te\_link\_flood\_scope\_set, [275](#)
- smi\_ospf\_te\_link\_flood\_scope\_-unset, [276](#)
- smi\_ospf\_te\_link\_set, [276](#)
- smi\_ospf\_telink\_te\_metric\_set, [277](#)
- smi\_ospf\_telink\_te\_metric\_unset, [277](#)
- smi\_ospf\_timers\_refresh\_set, [278](#)
- smi\_ospf\_timers\_refresh\_unset, [278](#)
- smi\_ospf\_timers\_spf\_set, [278](#)
- smi\_ospf\_timers\_spf\_unset, [279](#)
- smi\_ospf\_timers\_spf\_validate\_-and\_unset, [279](#)
- smi\_ospf\_vlink\_authentication\_-key\_set, [280](#)
- smi\_ospf\_vlink\_authentication\_-key\_unset, [281](#)

- smi\_ospf\_vlink\_authentication\_-  
type\_set, [281](#)
- smi\_ospf\_vlink\_authentication\_-  
type\_unset, [282](#)
- smi\_ospf\_vlink\_dead\_interval\_set,  
[282](#)
- smi\_ospf\_vlink\_dead\_interval\_-  
unset, [283](#)
- smi\_ospf\_vlink\_hello\_interval\_set,  
[283](#)
- smi\_ospf\_vlink\_hello\_interval\_-  
unset, [284](#)
- smi\_ospf\_vlink\_message\_digest\_-  
key\_set, [285](#)
- smi\_ospf\_vlink\_message\_digest\_-  
key\_unset, [285](#)
- smi\_ospf\_vlink\_retransmit\_-  
interval\_set, [286](#)
- smi\_ospf\_vlink\_retransmit\_-  
interval\_unset, [286](#)
- smi\_ospf\_vlink\_set, [287](#)
- smi\_ospf\_vlink\_transmit\_delay\_set,  
[287](#)
- smi\_ospf\_vlink\_transmit\_delay\_-  
unset, [288](#)
- smi\_ospf\_vlink\_unset, [289](#)
- smi\_show\_if\_info, [289](#)
- smi\_show\_ospf\_borderrouter\_info,  
[289](#)
- smi\_show\_ospf\_buffer\_info, [290](#)
- smi\_show\_ospf\_database\_detail\_-  
info, [290](#)
- smi\_show\_ospf\_database\_-  
summary\_info, [291](#)
- smi\_show\_ospf\_interface\_brief\_-  
info, [291](#)
- smi\_show\_ospf\_multiarea\_info, [292](#)
- smi\_show\_ospf\_nbr\_info, [292](#)
- smi\_show\_ospf\_proc\_info, [293](#)
- smi\_show\_ospf\_route\_info, [293](#)
- smi\_show\_ospf\_route\_summary\_-  
info, [293](#)
- smi\_show\_ospf\_vlink\_info, [294](#)
- smi\_ospf\_abr\_type\_set  
smi\_ospf.h, [72](#)
- smi\_ospf\_abr\_type\_unset  
smi\_ospf.h, [72](#)
- smi\_ospf\_area\_auth\_by\_type\_unset  
smi\_ospf.h, [73](#)
- smi\_ospf\_area\_auth\_type\_set  
smi\_ospf.h, [73](#)
- smi\_ospf\_area\_auth\_type\_unset  
smi\_ospf.h, [74](#)
- smi\_ospf\_area\_default\_cost\_set  
smi\_ospf.h, [74](#)
- smi\_ospf\_area\_default\_cost\_unset  
smi\_ospf.h, [75](#)
- smi\_ospf\_area\_default\_cost\_value\_unset  
smi\_ospf.h, [75](#)
- smi\_ospf\_area\_export\_list\_set  
smi\_ospf.h, [76](#)
- smi\_ospf\_area\_export\_list\_unset  
smi\_ospf.h, [76](#)
- smi\_ospf\_area\_filter\_list\_access\_set  
smi\_ospf.h, [77](#)
- smi\_ospf\_area\_filter\_list\_access\_unset  
smi\_ospf.h, [77](#)
- smi\_ospf\_area\_filter\_list\_prefix\_set  
smi\_ospf.h, [78](#)
- smi\_ospf\_area\_filter\_list\_prefix\_unset  
smi\_ospf.h, [78](#)
- smi\_ospf\_area\_import\_list\_set  
smi\_ospf.h, [79](#)
- smi\_ospf\_area\_import\_list\_unset  
smi\_ospf.h, [79](#)
- smi\_ospf\_area\_no\_summary\_set  
smi\_ospf.h, [80](#)
- smi\_ospf\_area\_no\_summary\_unset  
smi\_ospf.h, [80](#)
- smi\_ospf\_area\_nssa\_default\_originate\_-  
metric\_set  
smi\_ospf.h, [81](#)
- smi\_ospf\_area\_nssa\_default\_originate\_-  
metric\_type\_set  
smi\_ospf.h, [81](#)
- smi\_ospf\_area\_nssa\_default\_originate\_-  
route\_map\_set  
smi\_ospf.h, [82](#)
- smi\_ospf\_area\_nssa\_default\_originate\_-  
set  
smi\_ospf.h, [83](#)
- smi\_ospf\_area\_nssa\_default\_originate\_-  
unset  
smi\_ospf.h, [83](#)
- smi\_ospf\_area\_nssa\_no\_redistribution\_-  
set  
smi\_ospf.h, [84](#)
- smi\_ospf\_area\_nssa\_no\_redistribution\_-  
unset  
smi\_ospf.h, [84](#)

- smi\_ospf\_area\_nssa\_set
  - smi\_ospf.h, 85
- smi\_ospf\_area\_nssa\_stability\_interval\_-set
  - smi\_ospf.h, 85
- smi\_ospf\_area\_nssa\_translator\_role\_set
  - smi\_ospf.h, 86
- smi\_ospf\_area\_nssa\_translator\_role\_-unset
  - smi\_ospf.h, 86
- smi\_ospf\_area\_nssa\_unset
  - smi\_ospf.h, 87
- smi\_ospf\_area\_range\_not\_advertise\_set
  - smi\_ospf.h, 87
- smi\_ospf\_area\_range\_not\_advertise\_-unset
  - smi\_ospf.h, 88
- smi\_ospf\_area\_range\_set
  - smi\_ospf.h, 88
- smi\_ospf\_area\_range\_substitute\_set
  - smi\_ospf.h, 89
- smi\_ospf\_area\_range\_substitute\_unset
  - smi\_ospf.h, 90
- smi\_ospf\_area\_range\_unset
  - smi\_ospf.h, 90
- smi\_ospf\_area\_shortcut\_set
  - smi\_ospf.h, 91
- smi\_ospf\_area\_shortcut\_unset
  - smi\_ospf.h, 91
- smi\_ospf\_area\_stub\_set
  - smi\_ospf.h, 92
- smi\_ospf\_area\_stub\_unset
  - smi\_ospf.h, 92
- smi\_ospf\_auto\_cost\_reference\_-bandwidth\_set
  - smi\_ospf.h, 93
- smi\_ospf\_auto\_cost\_reference\_-bandwidth\_type\_set
  - smi\_ospf.h, 93
- smi\_ospf\_auto\_cost\_reference\_-bandwidth\_unset
  - smi\_ospf.h, 94
- smi\_ospf\_bfd.h, 295
  - smi\_ospf\_bfd\_all\_interfaces\_set, 296
  - smi\_ospf\_bfd\_all\_interfaces\_unset, 296
  - smi\_ospf\_if\_bfd\_disable\_set, 297
  - smi\_ospf\_if\_bfd\_disable\_unset, 297
  - smi\_ospf\_if\_bfd\_set, 297
  - smi\_ospf\_if\_bfd\_unset, 298
  - smi\_ospf\_vlink\_bfd\_set, 298
  - smi\_ospf\_vlink\_bfd\_unset, 299
- smi\_ospf\_bfd\_all\_interfaces\_set
  - smi\_ospf\_bfd.h, 296
- smi\_ospf\_bfd\_all\_interfaces\_unset
  - smi\_ospf\_bfd.h, 296
- smi\_ospf\_capability\_cspf\_set
  - smi\_ospf.h, 94
- smi\_ospf\_capability\_cspf\_unset
  - smi\_ospf.h, 95
- smi\_ospf\_capability\_opaque\_lsa\_set
  - smi\_ospf.h, 95
- smi\_ospf\_capability\_opaque\_lsa\_unset
  - smi\_ospf.h, 96
- smi\_ospf\_capability\_restart\_set
  - smi\_ospf.h, 96
- smi\_ospf\_capability\_restart\_unset
  - smi\_ospf.h, 96
- smi\_ospf\_capability\_traffic\_-engineering\_set
  - smi\_ospf.h, 97
- smi\_ospf\_capability\_traffic\_-engineering\_unset
  - smi\_ospf.h, 97
- smi\_ospf\_compatible\_rfc1583\_set
  - smi\_ospf.h, 98
- smi\_ospf\_compatible\_rfc1583\_unset
  - smi\_ospf.h, 98
- smi\_ospf\_cspf\_better\_protection\_type
  - smi\_ospf.h, 99
- smi\_ospf\_debug\_set
  - smi\_ospf.h, 99
- smi\_ospf\_debug\_unset
  - smi\_ospf.h, 101
- smi\_ospf\_default\_metric\_set
  - smi\_ospf.h, 102
- smi\_ospf\_default\_metric\_unset
  - smi\_ospf.h, 103
- smi\_ospf\_disable\_db\_summary\_opt
  - smi\_ospf.h, 103
- smi\_ospf\_disable\_ext\_multi\_inst
  - smi\_ospf.h, 103
- smi\_ospf\_distance\_all\_set
  - smi\_ospf.h, 104
- smi\_ospf\_distance\_all\_unset
  - smi\_ospf.h, 104
- smi\_ospf\_distance\_external\_set
  - smi\_ospf.h, 105
- smi\_ospf\_distance\_external\_unset

- smi\_ospf.h, 105
- smi\_ospf\_distance\_inter\_area\_set
  - smi\_ospf.h, 106
- smi\_ospf\_distance\_inter\_area\_unset
  - smi\_ospf.h, 106
- smi\_ospf\_distance\_intra\_area\_set
  - smi\_ospf.h, 107
- smi\_ospf\_distance\_intra\_area\_unset
  - smi\_ospf.h, 107
- smi\_ospf\_distance\_source\_set
  - smi\_ospf.h, 107
- smi\_ospf\_distance\_source\_unset
  - smi\_ospf.h, 108
- smi\_ospf\_distribute\_list\_in\_set
  - smi\_ospf.h, 109
- smi\_ospf\_distribute\_list\_in\_unset
  - smi\_ospf.h, 109
- smi\_ospf\_distribute\_list\_out\_set
  - smi\_ospf.h, 109
- smi\_ospf\_distribute\_list\_out\_unset
  - smi\_ospf.h, 110
- smi\_ospf\_dna\_set\_sdkapi
  - smi\_ospf.h, 111
- smi\_ospf\_dna\_unset\_sdkapi
  - smi\_ospf.h, 111
- smi\_ospf\_domain\_id\_set
  - smi\_ospf.h, 111
- smi\_ospf\_domain\_id\_unset
  - smi\_ospf.h, 112
- smi\_ospf\_enable\_db\_summary\_opt
  - smi\_ospf.h, 112
- smi\_ospf\_enable\_ext\_multi\_inst
  - smi\_ospf.h, 113
- smi\_ospf\_get\_address\_less\_if
  - smi\_ospf.h, 113
- smi\_ospf\_get\_admin\_stat
  - smi\_ospf.h, 114
- smi\_ospf\_get\_area\_aggregate\_area\_id
  - smi\_ospf.h, 114
- smi\_ospf\_get\_area\_aggregate\_effect
  - smi\_ospf.h, 115
- smi\_ospf\_get\_area\_aggregate\_lsdb\_type
  - smi\_ospf.h, 115
- smi\_ospf\_get\_area\_aggregate\_mask
  - smi\_ospf.h, 116
- smi\_ospf\_get\_area\_aggregate\_net
  - smi\_ospf.h, 116
- smi\_ospf\_get\_area\_aggregate\_route\_tag
  - smi\_ospf.h, 117
- smi\_ospf\_get\_area\_aggregate\_status
  - smi\_ospf.h, 117
- smi\_ospf\_get\_area\_bdr\_rtr\_count
  - smi\_ospf.h, 118
- smi\_ospf\_get\_area\_bdr\_rtr\_status
  - smi\_ospf.h, 118
- smi\_ospf\_get\_area\_id
  - smi\_ospf.h, 119
- smi\_ospf\_get\_area\_lsa\_cksum\_sum
  - smi\_ospf.h, 119
- smi\_ospf\_get\_area\_lsa\_count
  - smi\_ospf.h, 120
- smi\_ospf\_get\_area\_lsa\_count\_number
  - smi\_ospf.h, 120
- smi\_ospf\_get\_area\_nssa\_translator\_events
  - smi\_ospf.h, 120
- smi\_ospf\_get\_area\_nssa\_translator\_role
  - smi\_ospf.h, 121
- smi\_ospf\_get\_area\_nssa\_translator\_stability\_interval
  - smi\_ospf.h, 121
- smi\_ospf\_get\_area\_nssa\_translator\_state
  - smi\_ospf.h, 122
- smi\_ospf\_get\_area\_range\_area\_id
  - smi\_ospf.h, 122
- smi\_ospf\_get\_area\_range\_effect
  - smi\_ospf.h, 123
- smi\_ospf\_get\_area\_range\_mask
  - smi\_ospf.h, 123
- smi\_ospf\_get\_area\_range\_net
  - smi\_ospf.h, 124
- smi\_ospf\_get\_area\_range\_status
  - smi\_ospf.h, 124
- smi\_ospf\_get\_area\_status
  - smi\_ospf.h, 125
- smi\_ospf\_get\_area\_summary
  - smi\_ospf.h, 125
- smi\_ospf\_get\_as\_lsdb\_age
  - smi\_ospf.h, 125
- smi\_ospf\_get\_as\_lsdb\_checksum
  - smi\_ospf.h, 126
- smi\_ospf\_get\_as\_lsdb\_sequence
  - smi\_ospf.h, 126
- smi\_ospf\_get\_as\_scope\_lsa\_count
  - smi\_ospf.h, 127
- smi\_ospf\_get\_asbdr\_rtr\_count
  - smi\_ospf.h, 127
- smi\_ospf\_get\_asbdr\_rtr\_status
  - smi\_ospf.h, 128
- smi\_ospf\_get\_auth\_type

- smi\_ospf.h, [128](#)
- smi\_ospf\_get\_compatible\_rfc1583
  - smi\_ospf.h, [129](#)
- smi\_ospf\_get\_demand\_extensions
  - smi\_ospf.h, [129](#)
- smi\_ospf\_get\_discontinuity\_time
  - smi\_ospf.h, [129](#)
- smi\_ospf\_get\_domain\_id
  - smi\_ospf.h, [130](#)
- smi\_ospf\_get\_exit\_overflow\_interval
  - smi\_ospf.h, [130](#)
- smi\_ospf\_get\_ext\_lsdb\_advertisement
  - smi\_ospf.h, [131](#)
- smi\_ospf\_get\_ext\_lsdb\_age
  - smi\_ospf.h, [131](#)
- smi\_ospf\_get\_ext\_lsdb\_checksum
  - smi\_ospf.h, [132](#)
- smi\_ospf\_get\_ext\_lsdb\_limit
  - smi\_ospf.h, [132](#)
- smi\_ospf\_get\_ext\_lsdb\_lsid
  - smi\_ospf.h, [133](#)
- smi\_ospf\_get\_ext\_lsdb\_router\_id
  - smi\_ospf.h, [133](#)
- smi\_ospf\_get\_ext\_lsdb\_sequence
  - smi\_ospf.h, [134](#)
- smi\_ospf\_get\_ext\_lsdb\_type
  - smi\_ospf.h, [134](#)
- smi\_ospf\_get\_extern\_lsa\_cksum\_sum
  - smi\_ospf.h, [135](#)
- smi\_ospf\_get\_extern\_lsa\_count
  - smi\_ospf.h, [135](#)
- smi\_ospf\_get\_external\_type1\_metric
  - smi\_ospf.h, [136](#)
- smi\_ospf\_get\_external\_type2\_metric
  - smi\_ospf.h, [136](#)
- smi\_ospf\_get\_host\_area\_id
  - smi\_ospf.h, [136](#)
- smi\_ospf\_get\_host\_cfg\_area\_id
  - smi\_ospf.h, [137](#)
- smi\_ospf\_get\_host\_ip\_address
  - smi\_ospf.h, [137](#)
- smi\_ospf\_get\_host\_metric
  - smi\_ospf.h, [138](#)
- smi\_ospf\_get\_host\_status
  - smi\_ospf.h, [138](#)
- smi\_ospf\_get\_host\_tos
  - smi\_ospf.h, [139](#)
- smi\_ospf\_get\_if\_admin\_stat
  - smi\_ospf.h, [139](#)
- smi\_ospf\_get\_if\_area\_id
  - smi\_ospf.h, [140](#)
- smi\_ospf\_get\_if\_auth\_key
  - smi\_ospf.h, [140](#)
- smi\_ospf\_get\_if\_auth\_type
  - smi\_ospf.h, [141](#)
- smi\_ospf\_get\_if\_backup\_designated\_router
  - smi\_ospf.h, [141](#)
- smi\_ospf\_get\_if\_bdr
  - smi\_ospf.h, [142](#)
- smi\_ospf\_get\_if\_demand
  - smi\_ospf.h, [142](#)
- smi\_ospf\_get\_if\_designated\_router
  - smi\_ospf.h, [143](#)
- smi\_ospf\_get\_if\_dr
  - smi\_ospf.h, [143](#)
- smi\_ospf\_get\_if\_events
  - smi\_ospf.h, [143](#)
- smi\_ospf\_get\_if\_hello\_interval
  - smi\_ospf.h, [144](#)
- smi\_ospf\_get\_if\_ip\_address
  - smi\_ospf.h, [144](#)
- smi\_ospf\_get\_if\_lsa\_checksum
  - smi\_ospf.h, [145](#)
- smi\_ospf\_get\_if\_lsa\_count
  - smi\_ospf.h, [145](#)
- smi\_ospf\_get\_if\_metric\_address\_less\_if
  - smi\_ospf.h, [146](#)
- smi\_ospf\_get\_if\_metric\_ip\_address
  - smi\_ospf.h, [146](#)
- smi\_ospf\_get\_if\_metric\_status
  - smi\_ospf.h, [147](#)
- smi\_ospf\_get\_if\_metric\_value
  - smi\_ospf.h, [147](#)
- smi\_ospf\_get\_if\_multicast\_forwarding
  - smi\_ospf.h, [148](#)
- smi\_ospf\_get\_if\_poll\_interval
  - smi\_ospf.h, [148](#)
- smi\_ospf\_get\_if\_retrans\_interval
  - smi\_ospf.h, [149](#)
- smi\_ospf\_get\_if\_rtr\_dead\_interval
  - smi\_ospf.h, [149](#)
- smi\_ospf\_get\_if\_rtr\_priority
  - smi\_ospf.h, [150](#)
- smi\_ospf\_get\_if\_state
  - smi\_ospf.h, [150](#)
- smi\_ospf\_get\_if\_status
  - smi\_ospf.h, [151](#)
- smi\_ospf\_get\_if\_transit\_delay
  - smi\_ospf.h, [151](#)



- smi\_ospf\_get\_if\_type
  - smi\_ospf.h, [152](#)
- smi\_ospf\_get\_import\_as\_extern
  - smi\_ospf.h, [153](#)
- smi\_ospf\_get\_inter\_area\_metric
  - smi\_ospf.h, [153](#)
- smi\_ospf\_get\_intra\_area\_metric
  - smi\_ospf.h, [153](#)
- smi\_ospf\_get\_local\_lsdb\_age
  - smi\_ospf.h, [154](#)
- smi\_ospf\_get\_local\_lsdb\_checksum
  - smi\_ospf.h, [154](#)
- smi\_ospf\_get\_local\_lsdb\_sequence
  - smi\_ospf.h, [155](#)
- smi\_ospf\_get\_lsdb\_advertisement
  - smi\_ospf.h, [155](#)
- smi\_ospf\_get\_lsdb\_age
  - smi\_ospf.h, [156](#)
- smi\_ospf\_get\_lsdb\_area\_id
  - smi\_ospf.h, [156](#)
- smi\_ospf\_get\_lsdb\_checksum
  - smi\_ospf.h, [157](#)
- smi\_ospf\_get\_lsdb\_lsid
  - smi\_ospf.h, [158](#)
- smi\_ospf\_get\_lsdb\_router\_id
  - smi\_ospf.h, [158](#)
- smi\_ospf\_get\_lsdb\_sequence
  - smi\_ospf.h, [159](#)
- smi\_ospf\_get\_lsdb\_type
  - smi\_ospf.h, [159](#)
- smi\_ospf\_get\_nbma\_nbr\_permanence
  - smi\_ospf.h, [160](#)
- smi\_ospf\_get\_nbma\_nbr\_status
  - smi\_ospf.h, [160](#)
- smi\_ospf\_get\_nbr\_address\_less\_index
  - smi\_ospf.h, [161](#)
- smi\_ospf\_get\_nbr\_events
  - smi\_ospf.h, [161](#)
- smi\_ospf\_get\_nbr\_hello\_suppressed
  - smi\_ospf.h, [162](#)
- smi\_ospf\_get\_nbr\_ip\_addr
  - smi\_ospf.h, [162](#)
- smi\_ospf\_get\_nbr\_ls\_retrans\_qlen
  - smi\_ospf.h, [163](#)
- smi\_ospf\_get\_nbr\_options
  - smi\_ospf.h, [163](#)
- smi\_ospf\_get\_nbr\_priority
  - smi\_ospf.h, [164](#)
- smi\_ospf\_get\_nbr\_restart\_helper\_age
  - smi\_ospf.h, [164](#)
- smi\_ospf\_get\_nbr\_restart\_helper\_exit\_reason
  - smi\_ospf.h, [165](#)
- smi\_ospf\_get\_nbr\_restart\_helper\_status
  - smi\_ospf.h, [165](#)
- smi\_ospf\_get\_nbr\_rtr\_id
  - smi\_ospf.h, [166](#)
- smi\_ospf\_get\_nbr\_state
  - smi\_ospf.h, [166](#)
- smi\_ospf\_get\_opaque\_lsa\_support
  - smi\_ospf.h, [167](#)
- smi\_ospf\_get\_originate\_new\_lsas
  - smi\_ospf.h, [167](#)
- smi\_ospf\_get\_reference\_bandwidth
  - smi\_ospf.h, [168](#)
- smi\_ospf\_get\_restart\_age
  - smi\_ospf.h, [168](#)
- smi\_ospf\_get\_restart\_exit\_reason
  - smi\_ospf.h, [169](#)
- smi\_ospf\_get\_restart\_interval
  - smi\_ospf.h, [169](#)
- smi\_ospf\_get\_restart\_status
  - smi\_ospf.h, [169](#)
- smi\_ospf\_get\_restart\_strict\_lsa\_check
  - smi\_ospf.h, [170](#)
- smi\_ospf\_get\_restart\_support
  - smi\_ospf.h, [170](#)
- smi\_ospf\_get\_router\_id
  - smi\_ospf.h, [171](#)
- smi\_ospf\_get\_rx\_new\_lsas
  - smi\_ospf.h, [171](#)
- smi\_ospf\_get\_settrap
  - smi\_ospf.h, [171](#)
- smi\_ospf\_get\_spf\_runs
  - smi\_ospf.h, [172](#)
- smi\_ospf\_get\_stub\_area\_id
  - smi\_ospf.h, [172](#)
- smi\_ospf\_get\_stub\_metric
  - smi\_ospf.h, [173](#)
- smi\_ospf\_get\_stub\_metric\_type
  - smi\_ospf.h, [173](#)
- smi\_ospf\_get\_stub\_router\_advertisement
  - smi\_ospf.h, [174](#)
- smi\_ospf\_get\_stub\_router\_support
  - smi\_ospf.h, [174](#)
- smi\_ospf\_get\_stub\_status
  - smi\_ospf.h, [174](#)
- smi\_ospf\_get\_stub\_tos
  - smi\_ospf.h, [175](#)
- smi\_ospf\_get\_tos\_support

- smi\_ospf.h, [175](#)
- smi\_ospf\_get\_version\_number
  - smi\_ospf.h, [176](#)
- smi\_ospf\_get\_virt\_if\_area\_id
  - smi\_ospf.h, [176](#)
- smi\_ospf\_get\_virt\_if\_auth\_key
  - smi\_ospf.h, [177](#)
- smi\_ospf\_get\_virt\_if\_auth\_type
  - smi\_ospf.h, [177](#)
- smi\_ospf\_get\_virt\_if\_events
  - smi\_ospf.h, [177](#)
- smi\_ospf\_get\_virt\_if\_hello\_interval
  - smi\_ospf.h, [178](#)
- smi\_ospf\_get\_virt\_if\_lsa\_cksumsum
  - smi\_ospf.h, [178](#)
- smi\_ospf\_get\_virt\_if\_lsa\_count
  - smi\_ospf.h, [179](#)
- smi\_ospf\_get\_virt\_if\_neighbor
  - smi\_ospf.h, [179](#)
- smi\_ospf\_get\_virt\_if\_retrans\_interval
  - smi\_ospf.h, [180](#)
- smi\_ospf\_get\_virt\_if\_rtr\_dead\_interval
  - smi\_ospf.h, [180](#)
- smi\_ospf\_get\_virt\_if\_state
  - smi\_ospf.h, [181](#)
- smi\_ospf\_get\_virt\_if\_status
  - smi\_ospf.h, [181](#)
- smi\_ospf\_get\_virt\_if\_transit\_delay
  - smi\_ospf.h, [182](#)
- smi\_ospf\_get\_virt\_local\_lsdb\_age
  - smi\_ospf.h, [182](#)
- smi\_ospf\_get\_virt\_local\_lsdb\_checksum
  - smi\_ospf.h, [183](#)
- smi\_ospf\_get\_virt\_local\_lsdb\_sequence
  - smi\_ospf.h, [183](#)
- smi\_ospf\_get\_virt\_nbr\_area
  - smi\_ospf.h, [184](#)
- smi\_ospf\_get\_virt\_nbr\_events
  - smi\_ospf.h, [184](#)
- smi\_ospf\_get\_virt\_nbr\_hello\_suppressed
  - smi\_ospf.h, [185](#)
- smi\_ospf\_get\_virt\_nbr\_ip\_addr
  - smi\_ospf.h, [185](#)
- smi\_ospf\_get\_virt\_nbr\_ls\_retrans\_qlen
  - smi\_ospf.h, [186](#)
- smi\_ospf\_get\_virt\_nbr\_options
  - smi\_ospf.h, [186](#)
- smi\_ospf\_get\_virt\_nbr\_restart\_helper\_-age
  - smi\_ospf.h, [187](#)
- smi\_ospf\_get\_virt\_nbr\_restart\_helper\_-exit\_reason
  - smi\_ospf.h, [187](#)
- smi\_ospf\_get\_virt\_nbr\_restart\_helper\_-status
  - smi\_ospf.h, [188](#)
- smi\_ospf\_get\_virt\_nbr\_rtr\_id
  - smi\_ospf.h, [188](#)
- smi\_ospf\_get\_virt\_nbr\_state
  - smi\_ospf.h, [189](#)
- smi\_ospf\_graceful\_restart\_planned\_set\_-sdkapi
  - smi\_ospf.h, [189](#)
- smi\_ospf\_graceful\_restart\_planned\_-unset\_sdkapi
  - smi\_ospf.h, [190](#)
- smi\_ospf\_graceful\_restart\_set\_sdkapi
  - smi\_ospf.h, [190](#)
- smi\_ospf\_graceful\_restart\_unset\_sdkapi
  - smi\_ospf.h, [190](#)
- smi\_ospf\_host\_entry\_cost\_set
  - smi\_ospf.h, [191](#)
- smi\_ospf\_host\_entry\_cost\_unset
  - smi\_ospf.h, [191](#)
- smi\_ospf\_host\_entry\_set
  - smi\_ospf.h, [192](#)
- smi\_ospf\_host\_entry\_unset
  - smi\_ospf.h, [192](#)
- smi\_ospf\_if\_authentication\_key\_set
  - smi\_ospf.h, [193](#)
- smi\_ospf\_if\_authentication\_key\_set\_-by\_addr
  - smi\_ospf.h, [193](#)
- smi\_ospf\_if\_authentication\_key\_unset
  - smi\_ospf.h, [194](#)
- smi\_ospf\_if\_authentication\_key\_unset\_-by\_addr
  - smi\_ospf.h, [194](#)
- smi\_ospf\_if\_authentication\_type\_set
  - smi\_ospf.h, [195](#)
- smi\_ospf\_if\_authentication\_type\_set\_-by\_addr
  - smi\_ospf.h, [195](#)
- smi\_ospf\_if\_authentication\_type\_unset
  - smi\_ospf.h, [196](#)
- smi\_ospf\_if\_authentication\_type\_unset\_-by\_addr
  - smi\_ospf.h, [196](#)
- smi\_ospf\_if\_bfd\_disable\_set
  - smi\_ospf\_bfd.h, [297](#)

- smi\_ospf\_if\_bfd\_disable\_unset
  - smi\_ospf\_bfd.h, [297](#)
- smi\_ospf\_if\_bfd\_set
  - smi\_ospf\_bfd.h, [297](#)
- smi\_ospf\_if\_bfd\_unset
  - smi\_ospf\_bfd.h, [298](#)
- smi\_ospf\_if\_conf\_ldp\_igp\_set\_sdkapi
  - smi\_ospf.h, [196](#)
- smi\_ospf\_if\_conf\_ldp\_igp\_unset\_sdkapi
  - smi\_ospf.h, [197](#)
- smi\_ospf\_if\_cost\_set
  - smi\_ospf.h, [197](#)
- smi\_ospf\_if\_cost\_set\_by\_addr
  - smi\_ospf.h, [198](#)
- smi\_ospf\_if\_cost\_unset
  - smi\_ospf.h, [198](#)
- smi\_ospf\_if\_cost\_unset\_by\_addr
  - smi\_ospf.h, [198](#)
- smi\_ospf\_if\_cost\_value\_unset
  - smi\_ospf.h, [199](#)
- smi\_ospf\_if\_database\_filter\_set
  - smi\_ospf.h, [199](#)
- smi\_ospf\_if\_database\_filter\_set\_by\_addr
  - smi\_ospf.h, [200](#)
- smi\_ospf\_if\_database\_filter\_unset
  - smi\_ospf.h, [200](#)
- smi\_ospf\_if\_database\_filter\_unset\_by\_-  
addr
  - smi\_ospf.h, [200](#)
- smi\_ospf\_if\_dead\_interval\_set
  - smi\_ospf.h, [201](#)
- smi\_ospf\_if\_dead\_interval\_set\_by\_addr
  - smi\_ospf.h, [201](#)
- smi\_ospf\_if\_dead\_interval\_unset
  - smi\_ospf.h, [202](#)
- smi\_ospf\_if\_dead\_interval\_unset\_by\_-  
addr
  - smi\_ospf.h, [202](#)
- smi\_ospf\_if\_disable\_all\_set
  - smi\_ospf.h, [203](#)
- smi\_ospf\_if\_disable\_all\_unset
  - smi\_ospf.h, [203](#)
- smi\_ospf\_if\_dna\_set
  - smi\_ospf.h, [203](#)
- smi\_ospf\_if\_dna\_unset
  - smi\_ospf.h, [204](#)
- smi\_ospf\_if\_hello\_interval\_set
  - smi\_ospf.h, [204](#)
- smi\_ospf\_if\_hello\_interval\_set\_by\_addr
  - smi\_ospf.h, [205](#)
- smi\_ospf\_if\_hello\_interval\_unset
  - smi\_ospf.h, [205](#)
- smi\_ospf\_if\_hello\_interval\_unset\_by\_-  
addr
  - smi\_ospf.h, [206](#)
- smi\_ospf\_if\_ip\_router\_set
  - smi\_ospf.h, [206](#)
- smi\_ospf\_if\_ip\_router\_unset
  - smi\_ospf.h, [207](#)
- smi\_ospf\_if\_message\_digest\_key\_get
  - smi\_ospf.h, [207](#)
- smi\_ospf\_if\_message\_digest\_key\_set
  - smi\_ospf.h, [208](#)
- smi\_ospf\_if\_message\_digest\_key\_set\_-  
by\_addr
  - smi\_ospf.h, [208](#)
- smi\_ospf\_if\_message\_digest\_key\_unset
  - smi\_ospf.h, [209](#)
- smi\_ospf\_if\_message\_digest\_key\_-  
unset\_by\_addr
  - smi\_ospf.h, [209](#)
- smi\_ospf\_if\_mtu\_ignore\_set
  - smi\_ospf.h, [210](#)
- smi\_ospf\_if\_mtu\_ignore\_set\_by\_addr
  - smi\_ospf.h, [210](#)
- smi\_ospf\_if\_mtu\_ignore\_unset
  - smi\_ospf.h, [211](#)
- smi\_ospf\_if\_mtu\_ignore\_unset\_by\_addr
  - smi\_ospf.h, [211](#)
- smi\_ospf\_if\_mtu\_set
  - smi\_ospf.h, [211](#)
- smi\_ospf\_if\_mtu\_unset
  - smi\_ospf.h, [212](#)
- smi\_ospf\_if\_network\_p2mp\_nbma\_set
  - smi\_ospf.h, [212](#)
- smi\_ospf\_if\_network\_set
  - smi\_ospf.h, [213](#)
- smi\_ospf\_if\_network\_unset
  - smi\_ospf.h, [213](#)
- smi\_ospf\_if\_passive\_interface\_set
  - smi\_ospf.h, [213](#)
- smi\_ospf\_if\_passive\_interface\_unset
  - smi\_ospf.h, [214](#)
- smi\_ospf\_if\_priority\_set
  - smi\_ospf.h, [214](#)
- smi\_ospf\_if\_priority\_set\_by\_addr
  - smi\_ospf.h, [215](#)
- smi\_ospf\_if\_priority\_unset
  - smi\_ospf.h, [215](#)
- smi\_ospf\_if\_priority\_unset\_by\_addr

- smi\_ospf.h, [215](#)
- smi\_ospf\_if\_resync\_timeout\_set
  - smi\_ospf.h, [216](#)
- smi\_ospf\_if\_resync\_timeout\_set\_by\_-addr
  - smi\_ospf.h, [216](#)
- smi\_ospf\_if\_resync\_timeout\_unset
  - smi\_ospf.h, [217](#)
- smi\_ospf\_if\_resync\_timeout\_unset\_by\_-addr
  - smi\_ospf.h, [217](#)
- smi\_ospf\_if\_retransmit\_interval\_set
  - smi\_ospf.h, [218](#)
- smi\_ospf\_if\_retransmit\_interval\_set\_by\_addr
  - smi\_ospf.h, [218](#)
- smi\_ospf\_if\_retransmit\_interval\_unset
  - smi\_ospf.h, [218](#)
- smi\_ospf\_if\_retransmit\_interval\_unset\_by\_addr
  - smi\_ospf.h, [219](#)
- smi\_ospf\_if\_te\_metric\_set
  - smi\_ospf.h, [219](#)
- smi\_ospf\_if\_te\_metric\_unset
  - smi\_ospf.h, [220](#)
- smi\_ospf\_if\_transmit\_delay\_set
  - smi\_ospf.h, [220](#)
- smi\_ospf\_if\_transmit\_delay\_set\_by\_addr
  - smi\_ospf.h, [221](#)
- smi\_ospf\_if\_transmit\_delay\_unset
  - smi\_ospf.h, [221](#)
- smi\_ospf\_if\_transmit\_delay\_unset\_by\_-addr
  - smi\_ospf.h, [221](#)
- smi\_ospf\_log\_adj\_changes\_set
  - smi\_ospf.h, [222](#)
- smi\_ospf\_log\_adj\_changes\_unset
  - smi\_ospf.h, [222](#)
- smi\_ospf\_lsa\_min\_arrival\_set
  - smi\_ospf.h, [223](#)
- smi\_ospf\_lsa\_min\_arrival\_unset
  - smi\_ospf.h, [223](#)
- smi\_ospf\_lsa\_throttle\_timers\_set
  - smi\_ospf.h, [224](#)
- smi\_ospf\_lsa\_throttle\_timers\_unset
  - smi\_ospf.h, [224](#)
- smi\_ospf\_max\_area\_limit\_set\_sdkapi
  - smi\_ospf.h, [225](#)
- smi\_ospf\_max\_area\_limit\_unset\_sdkapi
  - smi\_ospf.h, [225](#)
- smi\_ospf\_max\_concurrent\_dd\_set
  - smi\_ospf.h, [225](#)
- smi\_ospf\_max\_concurrent\_dd\_unset
  - smi\_ospf.h, [226](#)
- smi\_ospf\_max\_unuse\_lsa\_set
  - smi\_ospf.h, [226](#)
- smi\_ospf\_max\_unuse\_lsa\_unset
  - smi\_ospf.h, [227](#)
- smi\_ospf\_max\_unuse\_packet\_set
  - smi\_ospf.h, [227](#)
- smi\_ospf\_max\_unuse\_packet\_unset
  - smi\_ospf.h, [227](#)
- smi\_ospf\_multi\_area\_adjacency\_set
  - smi\_ospf.h, [228](#)
- smi\_ospf\_multi\_area\_adjacency\_unset
  - smi\_ospf.h, [228](#)
- smi\_ospf\_nbr\_static\_cost\_set
  - smi\_ospf.h, [229](#)
- smi\_ospf\_nbr\_static\_cost\_unset
  - smi\_ospf.h, [229](#)
- smi\_ospf\_nbr\_static\_poll\_interval\_set
  - smi\_ospf.h, [230](#)
- smi\_ospf\_nbr\_static\_poll\_interval\_unset
  - smi\_ospf.h, [230](#)
- smi\_ospf\_nbr\_static\_priority\_set
  - smi\_ospf.h, [231](#)
- smi\_ospf\_nbr\_static\_priority\_unset
  - smi\_ospf.h, [231](#)
- smi\_ospf\_nbr\_static\_set
  - smi\_ospf.h, [232](#)
- smi\_ospf\_nbr\_static\_unset
  - smi\_ospf.h, [232](#)
- smi\_ospf\_network\_set
  - smi\_ospf.h, [233](#)
- smi\_ospf\_network\_unset
  - smi\_ospf.h, [234](#)
- smi\_ospf\_opaque\_area\_lsa\_set
  - smi\_ospf.h, [234](#)
- smi\_ospf\_opaque\_as\_lsa\_set
  - smi\_ospf.h, [235](#)
- smi\_ospf\_opaque\_link\_lsa\_set
  - smi\_ospf.h, [235](#)
- smi\_ospf\_opaque\_te\_link\_local\_lsa\_disable
  - smi\_ospf.h, [236](#)
- smi\_ospf\_opaque\_te\_link\_local\_lsa\_enable
  - smi\_ospf.h, [236](#)
- smi\_ospf\_overflow\_database\_external\_interval\_set

- smi\_ospf.h, [237](#)
- smi\_ospf\_overflow\_database\_external\_  
interval\_unset  
smi\_ospf.h, [237](#)
- smi\_ospf\_overflow\_database\_external\_  
limit\_set  
smi\_ospf.h, [238](#)
- smi\_ospf\_overflow\_database\_external\_  
limit\_unset  
smi\_ospf.h, [238](#)
- smi\_ospf\_passive\_interface\_default\_set  
smi\_ospf.h, [239](#)
- smi\_ospf\_passive\_interface\_default\_  
unset  
smi\_ospf.h, [239](#)
- smi\_ospf\_passive\_interface\_set  
smi\_ospf.h, [239](#)
- smi\_ospf\_passive\_interface\_set\_by\_addr  
smi\_ospf.h, [240](#)
- smi\_ospf\_passive\_interface\_unset  
smi\_ospf.h, [240](#)
- smi\_ospf\_passive\_interface\_unset\_by\_  
addr  
smi\_ospf.h, [241](#)
- smi\_ospf\_process\_set  
smi\_ospf.h, [241](#)
- smi\_ospf\_process\_set\_vrf  
smi\_ospf.h, [242](#)
- smi\_ospf\_process\_shut\_set  
smi\_ospf.h, [242](#)
- smi\_ospf\_process\_shut\_unset  
smi\_ospf.h, [243](#)
- smi\_ospf\_process\_unset  
smi\_ospf.h, [243](#)
- smi\_ospf\_redist\_default\_set  
smi\_ospf.h, [243](#)
- smi\_ospf\_redist\_default\_unset  
smi\_ospf.h, [244](#)
- smi\_ospf\_redist\_metric\_set  
smi\_ospf.h, [244](#)
- smi\_ospf\_redist\_metric\_type\_set  
smi\_ospf.h, [245](#)
- smi\_ospf\_redist\_metric\_type\_unset  
smi\_ospf.h, [246](#)
- smi\_ospf\_redist\_metric\_unset  
smi\_ospf.h, [246](#)
- smi\_ospf\_redist\_proto\_set  
smi\_ospf.h, [247](#)
- smi\_ospf\_redist\_proto\_unset  
smi\_ospf.h, [247](#)
- smi\_ospf\_redist\_tag\_set  
smi\_ospf.h, [248](#)
- smi\_ospf\_redist\_tag\_unset  
smi\_ospf.h, [248](#)
- smi\_ospf\_redistribute\_default\_set  
smi\_ospf.h, [249](#)
- smi\_ospf\_redistribute\_set  
smi\_ospf.h, [249](#)
- smi\_ospf\_restart\_graceful\_sdkapi  
smi\_ospf.h, [250](#)
- smi\_ospf\_restart\_helper\_grace\_period\_  
set  
smi\_ospf.h, [251](#)
- smi\_ospf\_restart\_helper\_grace\_period\_  
unset  
smi\_ospf.h, [251](#)
- smi\_ospf\_restart\_helper\_never\_router\_  
set  
smi\_ospf.h, [251](#)
- smi\_ospf\_restart\_helper\_never\_router\_  
unset  
smi\_ospf.h, [252](#)
- smi\_ospf\_restart\_helper\_never\_router\_  
unset\_all  
smi\_ospf.h, [252](#)
- smi\_ospf\_restart\_helper\_policy\_set  
smi\_ospf.h, [252](#)
- smi\_ospf\_restart\_helper\_policy\_unset  
smi\_ospf.h, [253](#)
- smi\_ospf\_routemap\_default\_set  
smi\_ospf.h, [253](#)
- smi\_ospf\_routemap\_default\_unset  
smi\_ospf.h, [254](#)
- smi\_ospf\_routemap\_set  
smi\_ospf.h, [254](#)
- smi\_ospf\_routemap\_unset  
smi\_ospf.h, [255](#)
- smi\_ospf\_router\_id\_set  
smi\_ospf.h, [255](#)
- smi\_ospf\_router\_id\_unset  
smi\_ospf.h, [256](#)
- smi\_ospf\_set\_area\_aggregate\_effect  
smi\_ospf.h, [256](#)
- smi\_ospf\_set\_area\_aggregate\_route\_tag  
smi\_ospf.h, [257](#)
- smi\_ospf\_set\_area\_aggregate\_status  
smi\_ospf.h, [257](#)
- smi\_ospf\_set\_area\_status  
smi\_ospf.h, [258](#)
- smi\_ospf\_set\_asbdr\_rtr\_status

- smi\_ospf.h, [258](#)
- smi\_ospf\_set\_if\_admin\_stat
  - smi\_ospf.h, [259](#)
- smi\_ospf\_set\_if\_area\_id
  - smi\_ospf.h, [259](#)
- smi\_ospf\_set\_if\_auth\_key
  - smi\_ospf.h, [260](#)
- smi\_ospf\_set\_if\_auth\_type
  - smi\_ospf.h, [260](#)
- smi\_ospf\_set\_if\_hello\_interval
  - smi\_ospf.h, [261](#)
- smi\_ospf\_set\_if\_metric\_status
  - smi\_ospf.h, [261](#)
- smi\_ospf\_set\_if\_metric\_value
  - smi\_ospf.h, [262](#)
- smi\_ospf\_set\_if\_poll\_interval
  - smi\_ospf.h, [262](#)
- smi\_ospf\_set\_if\_retrans\_interval
  - smi\_ospf.h, [263](#)
- smi\_ospf\_set\_if\_rtr\_dead\_interval
  - smi\_ospf.h, [264](#)
- smi\_ospf\_set\_if\_rtr\_priority
  - smi\_ospf.h, [264](#)
- smi\_ospf\_set\_if\_status
  - smi\_ospf.h, [265](#)
- smi\_ospf\_set\_if\_transit\_delay
  - smi\_ospf.h, [265](#)
- smi\_ospf\_set\_if\_type
  - smi\_ospf.h, [266](#)
- smi\_ospf\_set\_import\_as\_extern
  - smi\_ospf.h, [266](#)
- smi\_ospf\_set\_ldap\_limit\_sdkapi
  - smi\_ospf.h, [267](#)
- smi\_ospf\_set\_multicast\_extensions
  - smi\_ospf.h, [268](#)
- smi\_ospf\_set\_nbma\_nbr\_status
  - smi\_ospf.h, [268](#)
- smi\_ospf\_set\_nbr\_priority
  - smi\_ospf.h, [269](#)
- smi\_ospf\_set\_nssa\_stability\_interval
  - smi\_ospf.h, [269](#)
- smi\_ospf\_set\_settrap
  - smi\_ospf.h, [270](#)
- smi\_ospf\_set\_virt\_if\_retrans\_interval
  - smi\_ospf.h, [270](#)
- smi\_ospf\_set\_virt\_if\_status
  - smi\_ospf.h, [271](#)
- smi\_ospf\_summary\_address\_not\_-
  - advertise\_set
  - smi\_ospf.h, [271](#)
- smi\_ospf\_summary\_address\_not\_-
  - advertise\_unset
  - smi\_ospf.h, [272](#)
- smi\_ospf\_summary\_address\_set
  - smi\_ospf.h, [272](#)
- smi\_ospf\_summary\_address\_tag\_set
  - smi\_ospf.h, [273](#)
- smi\_ospf\_summary\_address\_tag\_unset
  - smi\_ospf.h, [273](#)
- smi\_ospf\_summary\_address\_unset
  - smi\_ospf.h, [274](#)
- smi\_ospf\_te\_link\_detail\_set
  - smi\_ospf.h, [274](#)
- smi\_ospf\_te\_link\_enable\_sdkapi
  - smi\_ospf.h, [275](#)
- smi\_ospf\_te\_link\_flood\_scope\_set
  - smi\_ospf.h, [275](#)
- smi\_ospf\_te\_link\_flood\_scope\_unset
  - smi\_ospf.h, [276](#)
- smi\_ospf\_te\_link\_set
  - smi\_ospf.h, [276](#)
- smi\_ospf\_telink\_te\_metric\_set
  - smi\_ospf.h, [277](#)
- smi\_ospf\_telink\_te\_metric\_unset
  - smi\_ospf.h, [277](#)
- smi\_ospf\_timers\_refresh\_set
  - smi\_ospf.h, [278](#)
- smi\_ospf\_timers\_refresh\_unset
  - smi\_ospf.h, [278](#)
- smi\_ospf\_timers\_spf\_set
  - smi\_ospf.h, [278](#)
- smi\_ospf\_timers\_spf\_unset
  - smi\_ospf.h, [279](#)
- smi\_ospf\_timers\_spf\_validate\_and\_unset
  - smi\_ospf.h, [279](#)
- smi\_ospf\_vlink\_authentication\_key\_set
  - smi\_ospf.h, [280](#)
- smi\_ospf\_vlink\_authentication\_key\_-
  - unset
  - smi\_ospf.h, [281](#)
- smi\_ospf\_vlink\_authentication\_type\_set
  - smi\_ospf.h, [281](#)
- smi\_ospf\_vlink\_authentication\_type\_-
  - unset
  - smi\_ospf.h, [282](#)
- smi\_ospf\_vlink\_bfd\_set
  - smi\_ospf\_bfd.h, [298](#)
- smi\_ospf\_vlink\_bfd\_unset
  - smi\_ospf\_bfd.h, [299](#)
- smi\_ospf\_vlink\_dead\_interval\_set

smi\_ospf.h, [282](#)  
smi\_ospf\_vlink\_dead\_interval\_unset  
smi\_ospf.h, [283](#)  
smi\_ospf\_vlink\_hello\_interval\_set  
smi\_ospf.h, [283](#)  
smi\_ospf\_vlink\_hello\_interval\_unset  
smi\_ospf.h, [284](#)  
smi\_ospf\_vlink\_message\_digest\_key\_set  
smi\_ospf.h, [285](#)  
smi\_ospf\_vlink\_message\_digest\_key\_-  
unset  
smi\_ospf.h, [285](#)  
smi\_ospf\_vlink\_retransmit\_interval\_set  
smi\_ospf.h, [286](#)  
smi\_ospf\_vlink\_retransmit\_interval\_-  
unset  
smi\_ospf.h, [286](#)  
smi\_ospf\_vlink\_set  
smi\_ospf.h, [287](#)  
smi\_ospf\_vlink\_transmit\_delay\_set  
smi\_ospf.h, [287](#)  
smi\_ospf\_vlink\_transmit\_delay\_unset  
smi\_ospf.h, [288](#)  
smi\_ospf\_vlink\_unset  
smi\_ospf.h, [289](#)  
smi\_show\_if\_info  
smi\_ospf.h, [289](#)  
smi\_show\_ospf\_borderrouter\_info  
smi\_ospf.h, [289](#)  
smi\_show\_ospf\_buffer\_info  
smi\_ospf.h, [290](#)  
smi\_show\_ospf\_database\_detail\_info  
smi\_ospf.h, [290](#)  
smi\_show\_ospf\_database\_summary\_info  
smi\_ospf.h, [291](#)  
smi\_show\_ospf\_interface\_brief\_info  
smi\_ospf.h, [291](#)  
smi\_show\_ospf\_multiarea\_info  
smi\_ospf.h, [292](#)  
smi\_show\_ospf\_nbr\_info  
smi\_ospf.h, [292](#)  
smi\_show\_ospf\_proc\_info  
smi\_ospf.h, [293](#)  
smi\_show\_ospf\_route\_info  
smi\_ospf.h, [293](#)  
smi\_show\_ospf\_route\_summary\_info  
smi\_ospf.h, [293](#)  
smi\_show\_ospf\_vlink\_info  
smi\_ospf.h, [294](#)