

SOLIDserver API: REST Reference Guide

Version 8.4



SOLIDserver API: REST Reference Guide

SOLIDserver API: REST Reference Guide

Revision: #141158

Publication date September 24, 2024

Copyright © 2000-2024 EfficientIP

All product specifications and information provided in this document are subject to change or update without notice and should not be construed as a commitment by EfficientIP. EfficientIP assumes no responsibility or liability for any mistakes, inaccuracies or omissions that may appear in this document. All statements and recommendations in this document are believed to be accurate at the time they are drafted but are presented without any representation or warranty of any kind, either express or implied, regarding their accuracy, completeness, performance, up-to-dateness or suitability for any particular use or purpose, or with respect to the infringement of any right. In particular, EfficientIP makes no representation or warranty that the results that may be obtained from your use of our products will be effective, accurate or reliable or that the quality of the products will meet your expectations. Users must take full responsibility for their application of any product.

This document aims at detailing EfficientIP proprietary solutions. As our solutions rely on several third-party products, created by other companies or organizations, it may redirect readers to third-party websites and documentation for further information. EfficientIP cannot be liable for or expected to provide said information regarding products maintained or created by third parties.

In no event shall EfficientIP be liable for any special, punitive, indirect, incidental or consequential damages of any kind including, but not limited to, loss of present or prospective profits or business, loss of data, business interruption, damages to reputation or image, whether in an action of contract, negligence, or other action, arising out of or in connection with the use, reliance upon or performance of the products provided by EfficientIP or any information contained herein.

All EfficientIP products and documentation are subject to separate licensing terms which users must agree to and comply with in order to use such products and documentation.

Table of Contents

| | |
|--|-----|
| About this Guide | xvi |
| Documentation Organization | xvi |
| Documentation Convention | xvi |
| I. REST Calls with SOLIDserver | 1 |
| 1. Technical Overview | 3 |
| Prerequisites | 3 |
| Limitations | 3 |
| 2. Calling SOLIDserver Services | 5 |
| Supported HTTP Verbs | 5 |
| REST Calls Format and Description | 5 |
| REST Calls Using Multiple Parameters | 6 |
| REST Calls Expected Response | 7 |
| 3. SOLIDserver Key Services | 8 |
| Services *_add | 8 |
| Services *_list | 8 |
| Services *_info | 10 |
| Services *_count | 11 |
| Services *_delete | 11 |
| 4. Calling Services With TAGS | 12 |
| Prerequisites | 12 |
| Limitations | 12 |
| Tagging a Class Parameter | 12 |
| Including Tagged Class Parameters in the Clause WHERE | 14 |
| Including Tagged Class Parameters in the Clause ORDERBY | 16 |
| Including Tagged Class Parameters in the Statements SELECT and GROUPBY | 18 |
| Expected Object Types in the TAGS | 20 |
| II. IPAM Services | 21 |
| 5. Space | 24 |
| ip_site_add | 25 |
| ip_site_list | 28 |
| ip_site_info | 31 |
| ip_site_count | 34 |
| group_site_add | 35 |
| group_site_delete | 37 |
| ip_site_delete | 39 |
| 6. IPv4 Network | 41 |
| ip_subnet_add | 42 |
| ip_block_subnet_list | 49 |
| ip_block_subnet_info | 56 |
| ip_block_subnet_count | 62 |
| ip_find_free_subnet | 63 |
| ip_block_subnet_groupby | 66 |
| ip_block_subnet_groupby_count | 68 |
| group_subnet_add | 70 |
| group_subnet_delete | 74 |
| ip_subnet_delete | 78 |
| 7. IPv6 Network | 82 |
| ip6_subnet6_add | 83 |
| ip6_block6_subnet6_list | 89 |
| ip6_block6_subnet6_info | 96 |

| | |
|--|-----|
| ip6_block6_subnet6_count | 102 |
| ip6_find_free_subnet6 | 103 |
| ip6_block6_subnet6_groupby | 105 |
| ip6_block6_subnet6_groupby_count | 107 |
| group_subnet6_add | 109 |
| group_subnet6_delete | 112 |
| ip6_subnet6_delete | 115 |
| 8. IPv4 Pool | 118 |
| ip_pool_add | 119 |
| ip_pool_list | 123 |
| ip_pool_info | 128 |
| ip_pool_count | 132 |
| group_pool_add | 133 |
| group_pool_delete | 136 |
| ip_pool_delete | 139 |
| 9. IPv6 Pool | 141 |
| ip6_pool6_add | 142 |
| ip6_pool6_list | 146 |
| ip6_pool6_info | 151 |
| ip6_pool6_count | 155 |
| group_pool6_add | 156 |
| group_pool6_delete | 159 |
| ip6_pool6_delete | 161 |
| 10. IPv4 Address | 163 |
| ip_add | 164 |
| ip_address_list | 169 |
| ip_address_info | 176 |
| ip_address_count | 183 |
| ip_find_free_address | 184 |
| ip_free_address_list | 187 |
| ip_used_address_list | 194 |
| ip_address_groupby | 201 |
| ip_address_groupby_count | 203 |
| ip_delete | 205 |
| 11. IPv6 Address | 207 |
| ip6_address6_add | 208 |
| ip6_address6_list | 212 |
| ip6_address6_info | 218 |
| ip6_address6_count | 224 |
| ip6_find_free_address6 | 225 |
| ip6_address6_groupby | 228 |
| ip6_address6_groupby_count | 230 |
| ip6_address6_delete | 232 |
| 12. IPv4 Address Alias | 234 |
| ip_alias_add | 235 |
| ip_alias_list | 238 |
| ip_alias_count | 241 |
| ip_alias_delete | 242 |
| 13. IPv6 Address Alias | 244 |
| ip6_alias_add | 245 |
| ip6_alias_list | 248 |
| ip6_alias_count | 251 |
| ip6_alias_delete | 252 |
| III. DHCP Services | 254 |

| | |
|-----------------------------------|-----|
| 14. DHCPv4 Server | 258 |
| dhcp_server_list | 259 |
| dhcp_server_info | 265 |
| dhcp_server_count | 271 |
| dhcp_server_options_list | 272 |
| 15. DHCPv6 Server | 275 |
| dhcp6_server6_list | 276 |
| dhcp6_server6_info | 281 |
| dhcp6_server6_count | 286 |
| dhcp6_server6_options6_list | 287 |
| 16. DHCPv4 Shared Network | 290 |
| dhcp_sn_add | 291 |
| dhcp_shared_network_list | 293 |
| dhcp_shared_network_info | 296 |
| dhcp_shared_network_count | 298 |
| dhcp_sn_delete | 299 |
| 17. DHCPv6 Shared Network | 301 |
| dhcp6_sn6_add | 302 |
| dhcp6_shared_network6_list | 304 |
| dhcp6_shared_network6_info | 307 |
| dhcp6_shared_network6_count | 309 |
| dhcp6_sn6_delete | 310 |
| 18. DHCPv4 Scope | 312 |
| dhcp_scope_add | 313 |
| dhcp_scope_list | 318 |
| dhcp_scope_info | 323 |
| dhcp_scope_count | 327 |
| dhcp_scope_groupby | 328 |
| dhcp_scope_groupby_count | 330 |
| dhcp_scope_options_list | 332 |
| group_dhcpscope_add | 335 |
| group_dhcpscope_delete | 338 |
| dhcp_scope_delete | 340 |
| 19. DHCPv6 Scope | 342 |
| dhcp6_scope6_add | 343 |
| dhcp6_scope6_list | 347 |
| dhcp6_scope6_info | 351 |
| dhcp6_scope6_count | 355 |
| dhcp6_scope6_options6_list | 356 |
| group_dhcpscope6_add | 359 |
| group_dhcpscope6_delete | 361 |
| dhcp6_scope6_delete | 363 |
| 20. DHCPv4 Group | 365 |
| dhcp_group_add | 366 |
| dhcp_group_list | 369 |
| dhcp_group_info | 373 |
| dhcp_group_count | 376 |
| dhcp_group_delete | 377 |
| 21. DHCPv6 Group | 379 |
| dhcp6_group6_list | 380 |
| 22. DHCPv4 Range | 383 |
| dhcp_range_add | 384 |
| dhcp_range_list | 388 |
| dhcp_range_info | 393 |

| | |
|--------------------------------------|-----|
| dhcp_range_count | 397 |
| dhcp_range_options_list | 398 |
| dhcp_range_delete | 401 |
| 23. DHCPv6 Range | 404 |
| dhcp6_range6_add | 405 |
| dhcp6_range6_list | 409 |
| dhcp6_range6_info | 414 |
| dhcp6_range6_count | 418 |
| dhcp6_range6_options6_list | 419 |
| dhcp6_range6_delete | 422 |
| 24. DHCPv4 Lease | 424 |
| dhcp_range_lease_list | 425 |
| dhcp_range_lease_info | 430 |
| dhcp_range_lease_count | 434 |
| dhcp_range_lease_groupby | 435 |
| dhcp_range_lease_groupby_count | 437 |
| dhcp_lease_log_list | 439 |
| dhcp_lease_log_count | 442 |
| dhcp_lease_log_groupby | 443 |
| dhcp_lease_log_groupby_count | 445 |
| dhcp_lease_manual_delete | 447 |
| 25. DHCPv6 Lease | 449 |
| dhcp6_lease6_list | 450 |
| dhcp6_lease6_count | 454 |
| dhcp6_lease6_log_list | 455 |
| dhcp6_lease6_log_count | 458 |
| dhcp6_lease6_log_groupby | 459 |
| dhcp6_lease6_log_groupby_count | 461 |
| 26. DHCPv4 Static | 463 |
| dhcp_static_add | 464 |
| dhcp_static_list | 469 |
| dhcp_static_info | 474 |
| dhcp_static_count | 479 |
| dhcp_static_groupby | 480 |
| dhcp_static_groupby_count | 482 |
| dhcp_static_options_list | 484 |
| dhcp_static_delete | 487 |
| 27. DHCPv6 Static | 490 |
| dhcp6_static6_add | 491 |
| dhcp6_static6_list | 495 |
| dhcp6_static6_info | 500 |
| dhcp6_static6_count | 504 |
| dhcp6_static6_options6_list | 505 |
| dhcp6_static6_delete | 508 |
| 28. DHCPv4 Option | 510 |
| dhcp_option_add | 511 |
| 29. DHCPv6 Option | 516 |
| dhcp6_option6_add | 517 |
| 30. DHCPv4 ACL and ACL Entry | 521 |
| dhcp_acl_add | 522 |
| dhcp_class_list | 525 |
| dhcp_class_info | 527 |
| dhcp_class_count | 529 |
| dhcp_acl_delete | 530 |

| | |
|--------------------------------------|-----|
| dhcp_acl_data_add | 532 |
| dhcp_subclass_list | 535 |
| dhcp_subclass_info | 538 |
| dhcp_subclass_count | 540 |
| dhcp_acl_data_delete | 541 |
| 31. DHCPv6 ACL and ACL Entry | 543 |
| dhcp6_acl6_add | 544 |
| dhcp6_class6_list | 547 |
| dhcp6_class6_info | 549 |
| dhcp6_class6_count | 551 |
| dhcp6_acl6_delete | 552 |
| dhcp6_acl_data6_add | 554 |
| dhcp6_subclass6_list | 557 |
| dhcp6_subclass6_info | 560 |
| dhcp6_subclass6_count | 562 |
| dhcp6_acl_data6_delete | 563 |
| 32. DHCPv4 Failover Channel | 565 |
| dhcp_failover_list | 566 |
| dhcp_failover_info | 569 |
| dhcp_failover_count | 572 |
| dhcp_failover_set_partner_down | 573 |
| IV. DNS Services | 575 |
| 33. DNS Server | 578 |
| dns_server_list | 579 |
| dns_server_info | 587 |
| dns_server_count | 595 |
| dns_smart_member_add | 596 |
| dns_smart_member_delete | 598 |
| 34. DNS View | 600 |
| dns_view_add | 601 |
| dns_view_list | 606 |
| dns_view_info | 610 |
| dns_view_count | 614 |
| group_dnsview_add | 615 |
| group_dnsview_delete | 617 |
| dns_view_delete | 619 |
| dns_view_param_add | 621 |
| dns_view_param_list | 623 |
| dns_view_param_info | 625 |
| dns_view_param_count | 627 |
| dns_view_param_delete | 628 |
| 35. DNS Zone | 630 |
| dns_zone_add | 631 |
| dns_zone_list | 639 |
| dns_zone_info | 647 |
| dns_zone_count | 654 |
| dns_zone_groupby | 655 |
| dns_zone_groupby_count | 657 |
| group_dnszone_add | 659 |
| group_dnszone_delete | 662 |
| dns_zone_delete | 665 |
| dns_zone_param_add | 667 |
| dns_zone_param_list | 669 |
| dns_zone_param_info | 671 |

| | |
|--|-----|
| dns_zone_param_count | 673 |
| dns_zone_param_delete | 674 |
| 36. DNS Resource Record | 676 |
| dns_rr_add | 677 |
| dns_rr_list | 686 |
| dns_rr_info | 695 |
| dns_rr_count | 703 |
| dns_rr_groupby | 704 |
| dns_rr_groupby_count | 706 |
| dns_rr_delete | 708 |
| 37. DNS ACL | 713 |
| dns_acl_add | 714 |
| dns_acl_list | 717 |
| dns_acl_info | 719 |
| dns_acl_count | 720 |
| dns_acl_delete | 721 |
| 38. TSIG Key | 723 |
| dns_key_add | 724 |
| dns_key_list | 727 |
| dns_key_info | 729 |
| dns_key_count | 731 |
| dns_key_delete | 732 |
| 39. DNSSEC | 734 |
| dnssec_zone_keys_list | 735 |
| dnssec_zone_keys_info | 738 |
| dnssec_enable_sign_zone | 740 |
| V. Network Object Manager Services | 745 |
| 40. Network Object Manager Folder | 747 |
| nom_folder_add | 748 |
| nom_folder_list | 752 |
| nom_folder_info | 755 |
| nom_folder_count | 757 |
| nom_folder_delete | 758 |
| 41. Network Object | 760 |
| nom_netobj_add | 761 |
| nom_netobj_list | 765 |
| nom_netobj_info | 768 |
| nom_netobj_count | 771 |
| nom_netobj_connected_list | 772 |
| nom_netobj_connected_count | 774 |
| nom_netobj_delete | 775 |
| 42. Interface | 777 |
| nom_iface_add | 778 |
| nom_iface_list | 782 |
| nom_iface_info | 786 |
| nom_iface_count | 789 |
| nom_port_list | 790 |
| nom_port_info | 792 |
| nom_port_count | 793 |
| nom_iface_delete | 794 |
| VI. Application Services | 797 |
| 43. Application | 799 |
| app_application_add | 800 |
| app_application_list | 803 |

| | |
|-------------------------------------|-----|
| app_application_info | 806 |
| app_application_count | 809 |
| app_application_groupby | 810 |
| app_application_groupby_count | 812 |
| app_application_delete | 814 |
| 44. Pool | 816 |
| app_pool_add | 817 |
| app_pool_list | 820 |
| app_pool_info | 823 |
| app_pool_count | 826 |
| app_pool_groupby | 827 |
| app_pool_groupby_count | 829 |
| app_pool_delete | 831 |
| 45. Node | 833 |
| app_node_add | 834 |
| app_node_list | 837 |
| app_node_info | 840 |
| app_node_count | 843 |
| app_node_groupby | 844 |
| app_node_groupby_count | 846 |
| app_node_delete | 848 |
| VII. Guardian Services | 850 |
| 46. Policy | 852 |
| guardian_policy_add | 853 |
| guardian_policy_list | 855 |
| guardian_policy_info | 857 |
| guardian_policy_count | 859 |
| guardian_policy_groupby | 860 |
| guardian_policy_groupby_count | 862 |
| guardian_policy_delete | 864 |
| VIII. Cloud Observer Services | 866 |
| 47. Plugin | 868 |
| co_plugin_list | 869 |
| 48. Worker | 871 |
| co_worker_add | 872 |
| co_worker_list | 876 |
| co_worker_info | 880 |
| co_worker_count | 883 |
| co_worker_delete | 884 |
| 49. Cloud Observer Folder | 886 |
| co_folder_list | 887 |
| co_folder_info | 890 |
| co_folder_count | 893 |
| 50. Cloud Observer Instance | 894 |
| co_instance_list | 895 |
| co_instance_info | 898 |
| co_instance_count | 901 |
| co_instance_network_list | 902 |
| co_instance_network_info | 906 |
| co_instance_network_count | 909 |
| 51. Cloud Observer Network | 910 |
| co_network_list | 911 |
| co_network_info | 914 |
| co_network_count | 917 |

| | |
|-------------------------------------|------|
| co_network_instance_list | 918 |
| co_network_instance_info | 922 |
| co_network_instance_count | 925 |
| 52. Cloud Observer IP Address | 926 |
| co_ip_list | 927 |
| co_ip_info | 931 |
| co_ip_count | 934 |
| co_ip_log_list | 935 |
| co_ip_log_info | 938 |
| co_ip_log_count | 940 |
| IX. NetChange Services | 941 |
| 53. Network Device | 943 |
| iplocator_netdev_add | 944 |
| iplnetdev_list | 948 |
| iplnetdev_info | 953 |
| iplnetdev_count | 957 |
| iplnetdev_groupby | 958 |
| iplnetdev_groupby_count | 960 |
| group_iplnetdev_add | 962 |
| group_iplnetdev_delete | 964 |
| iplocator_netdev_delete | 966 |
| 54. IPv4 Route | 968 |
| iplnetdevroute_list | 969 |
| iplnetdevroute_info | 974 |
| iplnetdevroute_count | 978 |
| iplnetdevroute_groupby | 979 |
| iplnetdevroute_groupby_count | 981 |
| 55. IPv6 Route | 983 |
| iplnetdevroute6_list | 984 |
| iplnetdevroute6_info | 989 |
| iplnetdevroute6_count | 993 |
| iplnetdevroute6_groupby | 994 |
| iplnetdevroute6_groupby_count | 996 |
| 56. NetChange VLAN | 998 |
| iplnetdevvlan_list | 999 |
| iplnetdevvlan_count | 1002 |
| iplnetdevvlan_groupby | 1003 |
| iplnetdevvlan_groupby_count | 1005 |
| 57. Port | 1007 |
| iplocator_port_add | 1008 |
| iplport_list | 1012 |
| iplport_info | 1018 |
| iplport_count | 1024 |
| 58. NetChange IPv4 Address | 1025 |
| iplnetdevaddr_list | 1026 |
| iplnetdevaddr_info | 1028 |
| iplnetdevaddr_count | 1030 |
| iplnetdevaddr_groupby | 1031 |
| iplnetdevaddr_groupby_count | 1033 |
| 59. NetChange IPv6 Address | 1035 |
| iplnetdevaddr6_list | 1036 |
| iplnetdevaddr6_info | 1038 |
| iplnetdevaddr6_count | 1040 |
| iplnetdevaddr6_groupby | 1041 |

| | |
|--------------------------------------|------|
| iplnetdevaddr6_groupby_count | 1043 |
| 60. Discovered Item | 1045 |
| ipldev_list | 1046 |
| ipldev_count | 1050 |
| ipldev_groupby | 1051 |
| ipldev_groupby_count | 1053 |
| ipldev_log_count | 1055 |
| ipldev_log_list | 1056 |
| X. Workflow Services | 1059 |
| 61. Request | 1061 |
| workflow_request_add | 1062 |
| request_incoming_list | 1067 |
| request_incoming_info | 1071 |
| request_incoming_count | 1074 |
| request_incoming_groupby | 1075 |
| request_incoming_groupby_count | 1077 |
| request_outgoing_list | 1079 |
| request_outgoing_info | 1083 |
| request_outgoing_count | 1086 |
| request_outgoing_groupby | 1087 |
| request_outgoing_groupby_count | 1089 |
| XI. Device Manager Services | 1091 |
| 62. Device Manager Device | 1093 |
| hostdev_add | 1094 |
| hostdev_list | 1098 |
| hostdev_info | 1101 |
| hostdev_count | 1103 |
| hostdev_groupby | 1104 |
| hostdev_groupby_count | 1106 |
| hostdev_delete | 1108 |
| 63. Port and Interface | 1110 |
| hostiface_add | 1111 |
| hostiface_list | 1115 |
| hostiface_info | 1119 |
| hostiface_count | 1122 |
| hostiface_groupby | 1123 |
| hostiface_groupby_count | 1125 |
| link_hostiface_add | 1127 |
| link_hostiface_list | 1130 |
| link_hostiface_count | 1132 |
| link_hostiface_delete | 1133 |
| hostiface_delete | 1135 |
| XII. VLAN Manager Services | 1137 |
| 64. VLAN Domain | 1139 |
| vlm_domain_add | 1140 |
| vlmdomain_list | 1143 |
| vlmdomain_info | 1145 |
| vlmdomain_count | 1147 |
| group_vlmdomain_add | 1148 |
| group_vlmdomain_delete | 1150 |
| vlm_domain_delete | 1152 |
| 65. VLAN Range | 1154 |
| vlm_range_add | 1155 |
| vlmrangelist | 1159 |

| | |
|--------------------------------------|------|
| vlmrange_info | 1162 |
| vlmrange_count | 1164 |
| group_vlmrange_add | 1165 |
| group_vlmrange_delete | 1167 |
| vlm_range_delete | 1169 |
| 66. VLAN | 1171 |
| vlm_vlan_add | 1172 |
| vlmvlan_list | 1176 |
| vlmvlan_info | 1179 |
| vlmvlan_count | 1182 |
| vlm_vlan_delete | 1183 |
| XIII. VRF Services | 1185 |
| 67. VRF | 1187 |
| vrf_vrfobject_add | 1188 |
| vrfobject_list | 1191 |
| vrfobject_info | 1193 |
| vrfobject_count | 1195 |
| vrf_vrfobject_delete | 1196 |
| 68. VRF Route Target | 1198 |
| vrf_linkvrfimportexport_add | 1199 |
| link_vrfimportexport_list | 1202 |
| vrf_linkvrfimportexport_delete | 1204 |
| XIV. Administration Services | 1206 |
| 69. Services Management | 1208 |
| service_list | 1209 |
| group_service_add | 1211 |
| group_service_delete | 1213 |
| 70. Group | 1215 |
| group_add | 1216 |
| group_list | 1220 |
| group_info | 1223 |
| group_count | 1225 |
| group_delete | 1226 |
| maintainer_group_list | 1227 |
| 71. User | 1229 |
| user_add | 1230 |
| user_info | 1235 |
| user_service_list | 1237 |
| group_service_list | 1239 |
| group_user_add | 1241 |
| group_user_delete | 1243 |
| user_delete | 1245 |
| 72. Custom Data | 1247 |
| custom_db_data_add | 1248 |
| custom_db_data_list | 1251 |
| custom_db_data_info | 1255 |
| custom_db_data_count | 1258 |
| custom_db_data_groupby | 1259 |
| custom_db_data_groupby_count | 1261 |
| custom_db_data_delete | 1263 |
| A. IPAM Cheat Sheet | 1265 |
| B. IPAM Workflow Sample | 1266 |
| C. DHCP Options | 1267 |
| Most Used Options | 1267 |

| | |
|-----------------------------|------|
| Basic | 1268 |
| Server Parameters | 1268 |
| Lease Information | 1269 |
| WINS/NetBIOS | 1270 |
| Host IP | 1270 |
| Interface | 1271 |
| Servers | 1272 |
| BootP Compatible | 1273 |
| DHCP Packet Fields | 1274 |
| Microsoft DHCP Client | 1275 |
| NetWare Client | 1276 |
| NIS/NISplus | 1276 |
| Miscellaneous | 1277 |
| Vendor Nwip | 1279 |
| Vendor MSFT | 1280 |
| D. Return Codes | 1281 |

List of Examples

| | |
|---|------|
| 3.1. Calling the service dhcp_static_add using Ruby | 8 |
| 3.2. Calling the service ip_block_subnet_list using PHP | 9 |
| 3.3. Calling the service ip_site_list using PHP and API token authentication | 9 |
| 3.4. Calling the service ip_site_list using a shell script and API token authentication | 10 |
| 3.5. Calling the service dns_rr_info using PHP | 10 |
| 3.6. Calling the service ip_address_count using Python | 11 |
| 3.7. Calling the service iplocator_netdev_delete using Python | 11 |
| 6.1. Calling the service ip_subnet_add using Python | 47 |
| 6.2. Calling the service ip_block_subnet_list using PHP, WHERE and ORDERBY | 55 |
| 6.3. Calling the service ip_find_free_subnet using Ruby | 65 |
| 6.4. Calling the service group_subnet_add using Python | 73 |
| 6.5. Calling the service ip_subnet_delete using PHP | 80 |
| 7.1. Calling the service ip6_block6_subnet6_list using Ruby and WHERE | 95 |
| 10.1. Calling the service ip_add using Ruby | 167 |
| 10.2. Calling the service ip_address_list using PHP | 175 |
| 10.3. Calling the service ip_address_info using PHP | 181 |
| 10.4. Calling the service ip_find_free_address using Python | 186 |
| 11.1. Calling the service ip6_address6_add using PHP | 211 |
| 11.2. Calling the service ip6_address6_delete using Ruby | 233 |
| 12.1. Calling the service ip_alias_list using Python and ORDERBY | 240 |
| 18.1. Calling the service dhcp_scope_list using PHP and WHERE | 322 |
| 24.1. Calling the service dhcp_range_lease_list using Python and WHERE | 429 |
| 26.1. Calling the service dhcp_static_add using Ruby | 468 |
| 26.2. Calling the service dhcp_static_info using Python | 478 |
| 30.1. Calling the service dhcp_subclass_list using PHP and ORDERBY | 537 |
| 31.1. Calling the service dhcp_subclass_list using PHP and ORDERBY | 558 |
| 35.1. Calling the service dns_zone_add using Ruby | 638 |
| 35.2. Calling the service dns_zone_list using Python | 645 |
| 36.1. Calling the service dns_rr_add using PHP | 685 |
| 36.2. Calling the service dns_rr_list using Python | 694 |
| 36.3. Calling the service dns_rr_info using Ruby | 702 |
| 36.4. Calling the service dns_rr_delete using PHP | 712 |
| 71.1. Calling the service user_add using PHP | 1234 |
| 71.2. Calling the service group_user_add using Python | 1242 |
| 71.3. Calling the service group_user_add using PHP | 1244 |
| 72.1. Calling the service custom_db_data_list using PowerShell | 1253 |

About this Guide

SOLIDserver can be managed using web services instead of, or in addition to, the GUI via REST mechanism. This guide provides an overview and description of the services you can execute.

Documentation Organization

This guide includes the following **parts** and **appendices**:

- [REST Calls with SOLIDserver](#): an introduction to REST mechanism, with a technical overview on how to make REST calls with SOLIDserver and a set of frequently asked questions.
- [IPAM Services](#): a part describing IP Address Management services, in IPv4 and IPv6, for spaces, networks, pools, IP addresses and IP address aliases.
- [DHCP Services](#): a part describing DHCP services for servers, shared networks, scopes, groups, ranges, leases, statics, options, ACLs and their entries and failover channels.
- [DNS Services](#): a part describing DNS services for servers, views, zones, resource records, ACLs, TSIG keys and DNSSEC.
- [Network Object Manager Services](#): a part describing Network Object Manager services for folders, network objects and interfaces.
- [Application Services](#): a part describing Application services for applications, pools and nodes.
- [Guardian Services](#): a part describing Guardian services for policies.
- [Cloud Observer Services](#): a part describing Cloud Observer services for plugins, workers, folders, instances, attached networks, networks, attached instances and IP addresses.
- [NetChange Services](#): a part describing NetChange services for network devices, IPv4 and IPv6 routes, VLANs, ports, NetChange IP addresses and discovered items.
- [Workflow Services](#): a part describing Workflow services for requests.
- [Device Manager Services](#): a part describing Device Manager services for devices and ports & interfaces.
- [VLAN Manager Services](#): a part describing VLAN Manager services for domains, ranges and VLANs.
- [VRF Services](#): a part describing VRF services for VRFs and VRF Route Targets.
- [Administration Services](#): a part describing some services of the module Administration that relate to services management, groups of users, users and custom data.
- [IPAM Cheat Sheet](#) an appendix describing key parameters when calling IPAM services.
- [IPAM Workflow Sample](#) an appendix including best practice scenarios to set up an IPAM services orchestration.
- [DHCP Options](#) an appendix describing the DHCP options.
- [Return Codes](#) an appendix listing all the return codes.

Documentation Convention

Each service is documented following the same structure, some keywords prevent confusions.

A Structured Description of each Service

Within the service dedicated parts, every service is detailed as follows:

| | |
|-------------|---------------------------------------|
| Name | The service name and general purpose. |
|-------------|---------------------------------------|

| | |
|-----------------------------------|---|
| Description | The detailed purpose of the service. |
| Mandatory Input Parameters | The combination(s) of parameters that you must use to call the service, if relevant. |
| Input Parameters | The description of all the available parameters to call the service. |
| Output Parameters | The description of all the parameters returned by the service once executed. This section does not include the parameters you may have tagged, for more details refer to the chapter Calling Services With TAGS . |

The Keywords in the Guide

Throughout the guide, we use the following keywords to avoid any confusion.

Child

An object that belongs to another object of same type in a VLSM organization.

For instance, a network can be the *child* of another network. A space can also be the *child* of another space.

Container

An object that can contain an object of another type. It can refer to an object at higher level within the hierarchy of the module.

For instance, the *container* of a DHCP static can be a server, scope or range.

Device

The highest level of the Device Manager hierarchy, it manages ports and interfaces. It can be a hardware appliance imported from NetChange, or a set of devices merged into one to ease the management of ports and interfaces. To avoid confusion, devices managed from this module are called **Device Manager devices** in this guide.

Database identifier

The identifier of an object within SOLIDserver database. The unique integer it is assigned once added to a table.

Folder

The highest level in Cloud Observer hierarchy. As they are also part of the Network Object Manager hierarchy, to avoid confusion, folders managed from this module are called **Cloud Observer folders** in this guide.

Folder

The highest level in Network Object Manager hierarchy. As they are also part of the Cloud Observer hierarchy, to avoid confusion, folders managed from this module are called **Network Object Manager folders** in this guide.

NetChange IPv4 or IPv6 Address

One of the second level objects of **NetChange** hierarchy. An IP address configured for an interface managed from a network device. All these addresses are listed on the dedicated page *All addresses* of the module.

Network device

A hardware appliance that was imported to SOLIDserver from the module NetChange. They can also be managed from the module Device Manager. To avoid any confusion, hardware appliances managed from NetChange are called **network devices** in this guide

Object

Any element that you can manage in SOLIDserver.

For instance, IPAM networks, DNS zones, NetChange ports, VLAN Manager domains, Network Object Manager network objects and Cloud Observer instances are all *objects*.

Parent

An object that can contain another object of the same type in a VLSM organization or in Network Object Manager.

In the IPAM, a non-terminal network can be the *parent* of another network and a space can also be the *parent* of another space.

In Network Object Manager, a folder can be the *parent* of another folder and a network object can also be the *parent* of another network object.

VLAN

The second level in **NetChange** hierarchy. Virtual Local Area Networks belong to network devices. They are called **NetChange VLAN** in this guide.

VLAN

The lowest level in **VLAN Manager** hierarchy. Virtual Local Area Networks can belong to VLAN ranges and/or VLAN domains. They can be associated with IPAM and DHCP objects. They are called **VLAN** in this guide.

Part I. REST Calls with SOLIDserver

Table of Contents

| | |
|--|----|
| 1. Technical Overview | 3 |
| Prerequisites | 3 |
| Limitations | 3 |
| 2. Calling SOLIDserver Services | 5 |
| Supported HTTP Verbs | 5 |
| REST Calls Format and Description | 5 |
| REST Calls Using Multiple Parameters | 6 |
| REST Calls Expected Response | 7 |
| 3. SOLIDserver Key Services | 8 |
| Services *_add | 8 |
| Services *_list | 8 |
| Services *_info | 10 |
| Services *_count | 11 |
| Services *_delete | 11 |
| 4. Calling Services With TAGS | 12 |
| Prerequisites | 12 |
| Limitations | 12 |
| Tagging a Class Parameter | 12 |
| Including Tagged Class Parameters in the Clause WHERE | 14 |
| Including Tagged Class Parameters in the Clause ORDERBY | 16 |
| Including Tagged Class Parameters in the Statements SELECT and GROUPBY | 18 |
| Expected Object Types in the TAGS | 20 |

Chapter 1. Technical Overview

The Representational State Transfer (**REST**) is a software architectural style used as an alternative to SOAP-based web services in web developments. It allows you to set up guidelines structured using HTTP verbs as constraints.

SOLIDserver provides a REST API based on basic HTTP/1.1 verbs mapped to CRUD operations (Create, Read, Update and Delete). You can use REST mechanism to execute SOLIDserver web services instead of, or in addition to, managing them through the graphical user interface. Almost all the operations available in the GUI can be performed using web services: additions, editing, deletions, retrieval of information, etc.

It presents the following **advantages**:

- REST allows the execution of CRUD services in a URL style.
- REST API format is supported in every language or software supporting HTTP.
- REST is suited for automating and scripting tasks.
- REST can be integrated within external applications.

Prerequisites

- SOLIDserver appliance must be configured with a correct hostname that can be resolved using a DNS query for the REST client.
- SOLIDserver appliance must respond to ping requests.
- The user must have the sufficient rights to execute the service.
- Every call requires user authentication, through:
 - Basic authentication.
 - The header, in base64 format.
 - An API token.
- The user must take into account the SSL certificate of their SOLIDserver session before executing each service. Depending on the configuration they may have to:
 - Accept the certificate from SOLIDserver GUI only once before executing web services.
 - Configure your calls execution to accept or ignore the certificate every time.

Limitations

- SOLIDserver supports the verbs *POST*, *GET*, *PUT*, *DELETE* and *OPTIONS*. For more details, refer to the chapter [Calling SOLIDserver Services](#).
- SOLIDserver does not support the HTTP verb *PATCH*.
- REST calls must respect the HTTP/1.1 format. Calls respecting HTTP/1.0 are not interpreted by the Apache server and may result in unpredicted behavior.
- REST calls apply to one object.
- REST allows you to **execute calls one by one**.

Therefore, any automated operation in the GUI has to be performed manually by calling all the relevant services one after the other. For instance, to delete from the GUI ten networks called **intranet** you would filter the page All networks, tick the networks and delete them all at once. With REST, you must call the service that lists the networks, retrieve the ID of the *intranet* networks, and then call the network deletion service ten times, for each network.

- You can no longer include class parameters in a clause *WHERE* using the structure `<object-type>_class_parameters like <value>`. To filter the results based on class parameters you must use TAGS instead, for more details refer to the chapter [Calling Services With TAGS](#).

Chapter 2. Calling SOLIDserver Services

To call web services SOLIDserver supports **5 HTTP verbs**, the execution of these calls can be more or less detailed and even **include a payload**.

Supported HTTP Verbs

Calling services via REST relies on the following HTTP verbs.

Table 2.1. Supported HTTP verbs

| Verb | Purpose | Used to call the service ^a |
|----------------|--|---------------------------------------|
| POST | Creating objects. Calls using that verb are not idempotent. | <i>*_add</i> |
| PUT | Updating/editing objects. Calls using that verb are idempotent ^b . | |
| GET | Retrieving object information. Calls using that verb are idempotent ^b . | <i>*_list, *_info and *_count</i> |
| DELETE | Deleting objects. Calls using that verb are idempotent ^b . | <i>*_delete</i> |
| OPTIONS | Retrieving a service help details. Calls using that verb are idempotent ^b . | any service |

^aThe services **_add*, **_list*, **_info*, **_count* and **_delete* are **SOLIDserver key services**, they exist for all objects.

^bCalling an idempotent service multiple times using the same verb and input parameters produces the same outcome, no matter how many times you call it.

Any other verb returns the HTTP error: *501 - Method Not Implemented*, *405 - Method Not Allowed* or *400 - Bad Request*.

REST Calls Format and Description

Every REST call must respect the following URL format.

`https://<your-SOLIDserver>/<rest-or-rpc>/<service-name>?[params-and-value]`

https://

Hypertext Transfer Protocol Secure is the only way to execute the service. `http://` returns the error 302.

<your-SOLIDserver>

The IP address or hostname used to connect to your appliance. The appliance must be running, you must have either imported the certificate or disabled the certificate validation. If you are using a web browser REST client, you should first acknowledge the warning message and ignore the self-signed certificate.

<rest-or-rpc>

The method used to execute the service and indicates the expected input parameters format.

`/rest/` must be used to call the services **_add*, **_list*, **_info*, **_count* and **_delete*.

`/rpc/` must be used to call any other service.

<service-name>

The name of the SOLIDserver service to be executed.

?

An optional separator. It is only required if you include input parameters in the call.

[params-and-value]

All the input parameter(s), if relevant for the service. Each parameter must be URL encoded and followed by its value, the expected format is detailed in the section [Multiple Parameters REST Calls Format](#).

The parameters order does not matter but this section of the URL is case sensitive, so make sure to indicate *WHERE* and not *where*, the same goes for *ORDERBY* or *SELECT*.

Keep in mind that **for each service**:

1. You must **execute the service with the appropriate HTTP verb**, the service returns an error otherwise.
2. You must **authenticate**, using the credentials of a user with sufficient rights and resources to execute the service.
 - You can either use **basic access authentication**.
 - Or you can **specify user credentials in the header**.

```
X-IPM-Username: <SOLIDserver-user-login-in-base64-format>
X-IPM-Password: <SOLIDserver-user-password-in-base64-format>
```

- Or you can **use an API token in the header**.

```
X-SDS-TS: <epoch-timestamp-in-seconds>
Authorization: SDS <Token-ID>:<StringToSign-in-SHA3-256-format>
```

With the StringToSign being:

```
<Token-Secret>
<epoch-timestamp-in-seconds>
<method>
<URL>
```

For examples, refer to [Calling a service using API token authentication](#).

3. You can indicate a **payload** using JSON format when relevant for the service. For more details, refer to the [REST calls with payload](#).

For calls without input parameters, to request a list for instance, you can use the format:

`https://<your-SOLIDserver>/<rest-or-rpc>/<service-name>`

If you use a REST client to execute our services, you can follow the procedure below.

To execute a service without input via a REST GUI client

1. Open your REST GUI client.
2. In the URL field type in `https://<your-SOLIDserver>/<rest-or-rpc>/<service-name>` .
3. Select the appropriate HTTP verb: *POST*, *GET*, *PUT*, *DELETE* or *OPTIONS*.
4. Specify your authentication credentials:
 - Either via basic access authentication;
 - Or in the header, using *X-IPM-Username: <base64-login>* and *X-IPM-Password: <base64-password>*, as detailed [above](#).
5. Send the request.

For calls with input parameters, we recommend using the format:

`https://<IP-address>/<rest-or-rpc>/<service-name>?<param>=<value>&....`
This format, the basic URI format to write down the URL, allows you to make sure your call is properly executed even if the value of an input parameter contains a "/".

REST Calls Using Multiple Parameters

Multiple parameters calls using REST can either include all the parameters and their value in the URL or include them in a payload, in the body of the call.

Multiple Parameter Calls via URL

No matter the service and parameters, a call via URL must respect the proper call format, detailed in the section [REST Calls Format and Description](#). To execute a service with input parameters via URL you must:

1. **Specify the parameters and their value.** There are two available formats:

- We recommend using `https://<your-SOLIDserver>/<rest-or-rpc>/<service-name>?<param1>=<URLencoded-param1-value>&<param2>=<URLencoded-param2-value>&...`

This format is especially useful if the value of any input parameter contains a "/".

- You can also use the deprecated format `https://<your-SOLIDserver>/<rest-or-rpc>/<service-name>/<param1>/<URLencoded-param1-value>/<param2>/<URLencoded-param2-value>/...`

2. **Respect HTTP/1.1 formatting.**

Do not hesitate to use a script that includes the service input parameters needed rather than converting every parameter value in URL encoding format. This script can also include several services as well.

Multiple Parameter Calls With Payload

Any service executed with **POST** or **PUT** can be sent with a request payload. It allows you to send a more detailed request for the service execution, with a specific ID, name or option for instance.

You cannot send a request payload for calls using **GET**, **DELETE** and **OPTIONS**.

To send a REST call with payload you must:

1. **Specify only the service in the URL** following the format: `https://<your-SOLIDserver>/<rest-or-rpc>/<service-name>`
2. **Respect HTTP/1.1 formatting.**
3. **Specify your payload in the body of the request with JSON formatted parameters and values.**

The payload can also be included in a script.

REST Calls Expected Response

- A service execution response is sent with an **HTTP status or error code**.
- The service returns the **output parameters in JSON format** in the body of the request.
- The **value of the output parameters returned can be URL encoded**.

Chapter 3. SOLIDserver Key Services

There are **five service types** that you can find in **all SOLIDserver modules**: `*_add`, `*_list`, `*_info`, `*_count` and `*_delete`; where `*` is the object the service applies to. Keep in mind that:

- These 5 service types **must be executed using the method `/rest/`** in the URL. For more details, refer to the section [REST Calls Format and Description](#).
- These 5 service types, except for `*_add` when used to create objects, are idempotent: calling them several times with the same parameter(s) specified in input does not change the output parameter(s) returned.
- **Any other type of service must be called using the method `/rpc/`.**

Services `*_add`

They allow you to **add** or **edit** objects in the database:

- Use the HTTP verb `POST` to add an object.
- Use the HTTP verb `PUT` and specify an existing identifier in input to edit an object.

Note that we recommend calling the services `*_add` with the input parameter **add_flag**. The value of this parameter allows you to overload the operation and make sure that you are either creating an object (`new_only`), or editing an existing object (`edit_only`).

The services `*_add` only apply to one object at a time. To add or edit several objects you must call the service as many times as there are objects involved.

Example 3.1. Calling the service `dhcp_static_add` using Ruby

```
require 'uri'
require 'net/http'

url = URI("https://solid.intranet/rest/dhcp_static_add?"+
          "dhcphost_mac_addr=01%3A0a%3A92%3Af2%3A54%3A17%3A80&dhcp_id=19")

http = Net::HTTP.new(url.host, url.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE

request = Net::HTTP::Post.new(url)
request["x-ipm-username"] = 'aXBtYWRtaW4='
request["x-ipm-password"] = 'YWRtaW4='
request["cache-control"] = 'no-cache'

response = http.request(request)
puts response.read_body
```

Services `*_list`

They allow you to retrieve the **list** of all the objects in a database.

Use the HTTP verb `GET` to call these services.

To filter or organize the results, you can specify the clause `WHERE`, the clause `ORDERBY`, the parameter `offset` and/or the parameter `limit` in input. The services `*_list` and `*_info` return the same parameters in output.

Example 3.2. Calling the service ip_block_subnet_list using PHP

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/ip_block_subnet_list",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Example 3.3. Calling the service ip_site_list using PHP and API token authentication

```
const _token = 'bd6f7316f093afafa7b9b64a31bfff8c9';
const _secret = '4731432dca7f4ba62079ac268326df4dcc05a8b880b79a500ba9054c68fa8188';

function signature($secret, $method, $url, $ts)
{
    $string = "{$secret}\n{$ts}\n{$method}\n{$url}";
    return hash('sha3-256', $string);
}

$ch = curl_init();

$method = "GET";
$url = "https://solid.intranet/rest/ip_site_list";

curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, $method);

$ts = time();
$token = _token;
$sig = signature(_secret, $method, $url, $ts);

$headers = []
$headers[] = "X-SDS-TS: {$ts}";
$headers[] = "Authorization: SDS {$token}:{$sig}";

curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

$result = curl_exec($ch);
```

```
print_r($result);
curl_close($ch);
```

Example 3.4. Calling the service ip_site_list using a shell script and API token authentication

```
EfficientIP_Token_Key="bd6f7316f093afafaf7b9b64a31bfff8c9"
EfficientIP_Token_Secret="4731432dca7f4ba62079ac268326df4dcc05a8b880b79a500ba9054c68fa8188"

URL="https://solid.intranet/rest/ip_site_list"
METHOD="GET"

TS=$(date +%s)

Sig=$(printf "$EfficientIP_Token_Secret\n$TS\n$METHOD\n$url" | openssl dgst -sha3-256 |
cut -d '=' -f 2 | tr -d ' ')

EfficientIP_CredsEncoded=$(printf "%b:%b" "$EfficientIP_Token_Key" "$Sig")
RET=$(curl -k -X $METHOD -H "Authorization: SDS $EfficientIP_CredsEncoded" -H "X-SDS-TS:
$TS" $url)
```

Services *_info

They allow you to retrieve the **properties** of a specific object.

Use the HTTP verb *GET* to call these services.

The services *_info only apply to one object at a time. They return the same parameters in output than the services *_list.

Example 3.5. Calling the service dns_rr_info using PHP

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/dns_rr_info?rr_id=204",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="),
));
$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Services * _count

They allow you to retrieve the **total number** of entries in the object database. This total includes all the objects: enabled or disabled, managed or unmanaged, in delayed create or delayed create. Only the objects that are already deleted from the database are excluded from the count.

Use the HTTP verb *GET* to call these services.

To filter the count result, you can specify the clause *WHERE* in input.

Example 3.6. Calling the service ip_address_count using Python

```
import requests

url = "https://solid.intranet/rest/ip_address_count"

headers = {
    'x-ipm-username': "aXBtYWRtaW4=",
    'x-ipm-password': "YWRtaW4=",
    'cache-control': "no-cache"
}

response = requests.request("GET", url, headers=headers)

print(response.text)
```

Services * _delete

They allow you to **delete** a specific object from the database.

Use the HTTP verb *DELETE* to call these services.

The services *_delete only apply to one object at a time. To delete several objects you must call the service as many times as there are objects involved.

Example 3.7. Calling the service iplocator_netdev_delete using Python

```
import requests

url = "https://solid.intranet/rest/iplocator_netdev_delete"

querystring = {"iplnetdev_id": "12"}

headers = {
    'x-ipm-username': "aXBtYWRtaW4=",
    'x-ipm-password': "YWRtaW4=",
    'cache-control': "no-cache"
}

response = requests.request("DELETE", url, headers=headers, params=querystring)

print(response.text)
```

Chapter 4. Calling Services With TAGS

TAGS is a proprietary tagging attribute designed to **retrieve the class parameters** - i.e. custom class parameters, advanced properties or metadata - configured on an object.

Tagging a class parameter allows you to find it more easily in the output parameters of a service or to use it in the clauses and statements **WHERE**, **ORDERBY**, **SELECT** and **GROUPBY**.

All the class parameters and their value, are concatenated and separated by a comma in the value of the parameter `<object-type>_class_parameters`. Using **TAGS** returns an additional parameter called `tag_<your-class-parameter>` that allows you to single out a class parameter information on a dedicated line; the class parameter is tagged.

Since version 6.0.0, it is no longer possible to include in a clause **WHERE** the structure `<object-type>_class_parameters like <value>`, you must tag the class parameter of your choice and then include it in the clause.

Prerequisites

- Calling a service for an object configured with custom class parameters, advanced properties or metadata.
- Specifying the proper object type, as detailed in the table [Expected object types in the TAGS](#).
- Encoding the calls. All calls using TAGS must be URL encoded.

Limitations

- TAGS can only be used on services `*_list` and `*_info`.
- TAGS can only be used for calls that return the parameter `<object-type>_class_parameters`.
- TAGS cannot single out the inheritance/propagation properties of a class parameter. It cannot retrieve the value of the parameters `<object-type>_class_parameters_properties` and `<object-type>_class_parameters_inheritance_source`.

Tagging a Class Parameter

To tag a class parameter, you must respect the following URL format.

`https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.<param>`

All calls must be URL encoded, but, as tagging one class parameter only includes a . (dot), the call does not need encoding.

`https://<your-SOLIDserver>/rest/<service-name>?`

For more details, refer to the section [REST Calls Format and Description](#).

TAGS=

Specifies that you want to extract a specific class parameter from the output parameter `<object-type>_class_parameters`.

<object-type>.<param>

The expected name convention to tag the class parameter and add the extra line in the output. `<object-type>` and `<param>` must be separated with a . (dot).

`<object-type>` is the type of object the class parameter applies to. All types are listed in the table [Expected object types in the TAGS](#).

<param> is the name of the class parameter you want to return independently.

In the following example, we use TAGS to call the service *dhcp_scope_list* and retrieve a class parameter called *information*. The URL below cannot be used as such, you must execute the call respecting the format detailed in the section [REST Calls Format and Description](#).

```
https://192.168.0.1/rest/dhcp_scope_list?TAGS=dhcpscope.information
```

The service returns all the output parameters, plus a dedicated line for the tagged class parameter:

```
{
    "errno": "0",
    "vdhcp_parent_id": "0",
    "vdhcp_arch": "splitscope",
    "dhcp_type": "vdhcp",
    "dhcpfailover_id": "13",
    "dhcpfailover_name": "failover-mycompany.corporation",
    "dhcpscope_id": "951",
    "dhcp_id": "19",
    "dhcp_name": "mycompany.corporation",
    "dhcpscope_name": "allocation",
    "dhcpscope_start_ip_addr": "0d000000",
    "dhcpscope_end_ip_addr": "0xffffffff",
    "dhcpscope_net_addr": "13.0.0.0",
    "dhcpscope_net_mask": "255.0.0.0",
    "dhcpscope_size": "16777216",
    "delayed_create_time": "0",
    "delayed_delete_time": "0",
    "dhcpscope_site_name": "#",
    "dhcpscope_site_id": "0",
    "dhcpscope_sort_name": "",
    "dhcpscope_class_name": "info",
    "dhcpsn_id": "949",
    "dhcpsn_name": "13.0.0.0/8",
    "vdhcp_parent_name": "#",
    "dhcp_class_name": "",
    "dhcp_version": "",
    "row_enabled": "1",
    "ip_addr": "#",
    "multistatus": "",
    "tag_dhcpscope_information": "important data",
        <!--The class parameter you tagged is also returned by
<object-type>.class_parameters-->
    "dhcpscope_class_parameters": "ipam_replication=0&information=important%20data",
    "dhcpscope_class_parameters_properties":
"ipam_replication=inherited,propagate&information=set,propagate",
    "dhcpscope_class_parameters_inheritance_source":
"ipam_replication=real_dhcp,19&information=real_dhcpscope,951",
    "dhcp_class_parameters": "ipam_replication=0",
    "dhcpscope_class_parameters_properties": "ipam_replication=set,propagate"
}
```

Keep in mind that you can tag several class parameters in one call. The call format to tag two or more class parameters is the following:

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type1>.<param1>&<object-type2>.<param2>&...
```

The URL encoded version of that call is the following:

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type1>.<param1>%26<object-type2>.<param2>%26...
```

Including Tagged Class Parameters in the Clause WHERE

As it is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`, tagging class parameters is the only way to include them in the filtering clause `WHERE`.

Filtering Results based on One Class Parameter

To filter the service result based on a class parameter, you must tag the class parameter first and then include the tagged parameter in the clause `WHERE` following the URL format:

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.〈param>&WHERE=tag_<object-type>_<param> <filter>
```

Once URL encoded, the call format is the following:

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.〈param>&WHERE=tag_<object-type>_<param>%20<filter>
```

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.〈param>
```

For more details, refer to the section [Tagging a Class Parameter](#).

&WHERE=

The clause that filters the results of the service. It must include the full name of the tagged class parameter.

tag_<object-type>_<param>

The name of the tagged class parameter as returned by the service. It always starts with `tag_`. It is followed by the object type, all types are listed in the table [Expected object types in the TAGS](#). Finally, the class parameter *Name* is mentioned, not its *Label*.

<filter>

The value of the class parameter that you want to filter the result with.

Use the structure `tag_<object-type>_<param> like '<param-value>'` to specify a string value. To indicate that the class parameter should contain the `<param-value>` but that it can contain other characters, you can use the character `%` as a wildcard: `'%<param-value>%'`. The space must be encoded: `tag_<object-type>_<param>%20like%20%27<param-value>%27`. The code for the `%` is `%25`.

Use the structure `tag_<object-type>_<param>='<param-value>'` to specify an integer value. You can also use other operators like `>`, `>=`, `<` ... Within the clause, `=` must be encoded: `tag_<object-type>_<param>%3D%27<param-value>%27`.

In the following example, we call the service `dhcp_scope_list` using TAGS and the clause `WHERE` to only return scopes configured with the class parameter *information* and the value *important*. The URL below cannot be used as such, you must execute the call respecting the format detailed in the section [REST Calls Format and Description](#).

```
https://192.168.0.1/rest/dhcp_scope_list?TAGS=dhcpscope.information&WHERE=tag_dhcpscope_information%20like%20%27important%27  
// In readable format the clause WHERE contains: tag_dhcpscope_information like 'important'
```

Filtering Results based on Multiple Class Parameters

To filter based on multiple class parameters, you must tag all the class parameters and specify the value of each parameter in the clause *WHERE*. The expected format is the following:

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type1>.<param1>&<object-type2>.<param2>&WHERE=tag_<object-type1>_<param1> like '<param1-value>' <and-or-or> tag_<object-type2>_<param2>='<param2-value>'
```

Once URL encoded, the call format is the following:

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type1>.<param1>%26<object-type2>.<param2>&WHERE=tag_<object-type1>_<param1>%20like%20%27<param1-value>%27 %20<and-or-or>%20tag_<object-type2>_<param2>%3D%27<param2-value>%27
```

https://<your-SOLIDserver>/rest/<service-name>?TAGS=

For more details, refer to the section [Tagging a Class Parameter](#).

<object-type1>.<param1>&<object-type2>.<param2>

The list of all the class parameters you want to tag and use in the clause (**<object-type>.<param>**), separated by a **&**, you can add as many class parameters as you need.

Within the clause, **&** must be encoded:**<object-type1>.<param1>%26<object-type2>.<param2>**.

&WHERE=

The clause that filters the service output parameters. It must include the full name of all the tagged class parameters.

tag_<object-type1>_<param1> like '<param1-value>'

The name of the first tagged class parameter (**tag_<object-type>_<param>**) you want to filter. For a string value use the encoded version of *like '<param1-value>'*.

%20<and-or-or>%20

The condition in which you want to filter the results. You can include all class parameters (**and**) or either class parameter (**or**) in the output parameters.

tag_<object-type2>_<param2>='<param2-value>'

The name of the second tagged class parameter (**tag_<object-type>_<param>**) you want to filter. For an integer use the encoded version of **=<param2-value>**. You can add more, as long as it is preceded by **<and-or-or>**.

In the following example, we call the service *dhcp_scope_list* using TAGS and the clause *WHERE* to filter the result and only return either scopes which class parameter *information* is *important* or scopes which class parameter *description* contains *accounting*. The URL below cannot be used as such, you must execute the call respecting the format detailed in the section [REST Calls Format and Description](#).

```
https://192.168.0.1/rest/dhcp_scope_list?TAGS=dhcpscope.information%26dhcpscope.description&WHERE=tag_dhcpscope_information%20like%20%27important%27%20or%20tag_dhcpscope_description%20like%20%27%25accounting%25%27  
// In readable format the clause WHERE contains: tag_dhcpscope_information like 'important' or tag_dhcpscope_description like '%accounting%'
```

Including Tagged Class Parameters in the Clause ORDERBY

Tagging class parameters is the only way to include them in the clause *ORDERBY* to sort the results.

Keep in mind that the order of the parameters specified in the clause is set using their value (name or ordinal number). If a parameter has the same value in two different rows, the next parameter of each row is compared. If that second parameter is still the same for both rows, the next parameter of each row is compared, and so on until all the rows are ordered according to the configuration you set in the clause. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

Sorting Results based on One Class Parameter

To sort the service result based on one class parameter, you must tag the class parameter first and then include the tagged parameter in the clause *ORDERBY* as follows.

`https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.〈param〉&ORDERBY=tag_<object-type>_〈param〉 <order>`

Once URL encoded, the call format is the following:

`https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.〈param〉&ORDERBY=tag_<object-type>_〈param〉%20<order>`

`https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.〈param〉`

For more details, refer to the section [Tagging a Class Parameter](#).

&ORDERBY=

The clause that indicates that the output parameters must be sorted.

tag_<object-type>_〈param〉

The name of the tagged class parameter as returned by the service. It always starts with **tag_**. It is followed by the object type, all types are listed in the table [Expected object types in the TAGS](#). Finally, the class parameter *Name* is mentioned, not its *Label*.

<order>

The order in which you want the results to be returned, based on the value of the class parameter **tag_<object-type>_〈param〉**. It can be either **ASC** (ascending) or **DESC** (descending) and preceded by an encoded space: **tag_<object-type>_〈param〉%20<ASC-or-DESC>**.

The **<order>** is optional, if you do not specify an order, the service returns information in ascending order.

In the following example, we call the service *dhcp_scope_list* using TAGS and the clause *ORDERBY* to sort the scopes based on the value of the class parameter *information*. The URL below cannot be used as such, you must execute the call respecting the format detailed in the section [REST Calls Format and Description](#).

```
https://192.168.0.1/rest/dhcp_scope_list?TAGS=dhcpscope.information&ORDERBY=tag_dhcpscope_information%20DESC
// In readable format the clause ORDERBY contains: tag_dhcpscope_information DESC
```

Sorting Results based on Multiple Class Parameter

To sort the service result based on multiple class parameters, you must tag all the class parameters and specify the value of each parameter in the clause *ORDERBY* as follows.

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type1>.<param1>&<object-type2>.<param2>&ORDERBY=tag_<object-type1>_<param1> <order>, tag_<object-type2>_<param2> <order>
```

Once URL encoded, the call format is the following:

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type1>.<param1>%26<object-type2>.<param2>&ORDERBY=tag_<object-type1>_<param1>%20<order>%2C%20tag_<object-type2>_<param2>%20<order>
```

https://<your-SOLIDserver>/rest/<service-name>?TAGS=

For more details, refer to the section [Tagging a Class Parameter](#).

<object-type1>.<param1>&<object-type2>.<param2>

The list of all the class parameters you want to tag and use in the clause (**<object-type>.<param>**), separated by a **&**, you can add as many class parameters as you need. Within the clause, **&** must be encoded: **<object-type1>.<param1>%26<object-type2>.<param2>**.

&ORDERBY=

The clause that indicates that the output parameters must be sorted. It must include the full name of all the tagged class parameters, you can add as many as you want.

tag_<object-type1>_<param1> <order>

The name of the first tagged class parameter (**tag_<object-type1>_<param1>**) you want to sort the result with. You can specify a sorting **<order>**: **ASC** (ascending) or **DESC** (descending). The tagged class parameter and order must be separated by an encoded space: **tag_<object-type>_<param>%20<ASC-or-DESC>**. Without order, the results are returned in ascending order.

,

The required separator between tagged class parameters. You must insert it after the **tag_<object-type1>_<param1>** or the **<order>**. It must be encoded **%2C**.

tag_<object-type2>_<param2> <order>

The name of the second tagged class parameter (**tag_<object-type2>_<param2>**) that sorts the result. You can specify a sorting **<order>** for that one too, separate the tagged class parameter and order with a space. Without order, the results are returned in ascending order. You can add more, as long as it is preceded by an encoded comma: **%2C**.

In the example below, we call the service *dhcp_scope_list* using TAGS to sort the scopes based first on the value of the class parameter *information*, in ascending order, and then on the value of the class parameter *description*, in descending order:

```
https://192.168.0.1/rest/dhcp_scope_list?TAGS=dhcpscope.information%26dhcpscope.description&ORDERBY=tag_dhcpscope_information%20%2Ctag_dhcpscope_description%20DESC  
  
// In readable format the clause ORDERBY contains: tag_dhcpscope_information, tag_dhcpscope_description DESC
```

Including Tagged Class Parameters in the Statements SELECT and GROUPBY

Tagging class parameters is the only way to include them in the statements *SELECT* and *GROUPBY* of the services *_groupby and *_groupby_count.

Aggregating Results based on Class Parameters

To aggregate the service result based on class parameters, you must tag the class parameters first and then include the tagged parameter in the statements *SELECT* and *GROUPBY* as follows.

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.<param>&SELECT=tag_<object-type>_<param>&GROUPBY=tag_<object-type>_<param>
```

All calls must be URL encoded, but, as tagging one class parameter only includes a . (dot), the call does not need encoding if you do not use aggregation functions in either statement.

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.<param>
```

For more details, refer to the section [Tagging a Class Parameter](#).

&SELECT=

The statement that indicates which parameter is returned in output. You can include an aggregation function in this statement: *count*, *max*, *min*, *sum* or *avg*.

You can specify several class parameters. The order of the specified parameters is respected in output, all parameters must be separated by a comma, encoded as follows %2C .

tag_<object-type>_<param>

The name of the tagged class parameter as returned by the service. It always starts with **tag_**. It is followed by the object type, all types are listed in the table [Expected object types in the TAGS](#). Finally, the class parameter *Name* is mentioned, not its *Label*.

In the example below, we use the aggregation function *count*, it must precede the parameter as follows: *count(<parameter>)*. To count any parameter specified in the statement, you can use *count(*)*, *<parameter>* .

&GROUPBY=

The statement that aggregates the results using the output parameter specified in this statement or in the statement *SELECT*. Keep in mind that any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

You can specify several class parameters. The order of the specified parameters is respected in output, all parameters must be separated by a comma, encoded as follows %2C . Keep in mind that any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

tag_<object-type>_<param>

The name of the tagged class parameter as returned by the service. It always starts with **tag_**. It is followed by the object type, all types are listed in the table [Expected object types in the TAGS](#). Finally, the class parameter *Name* is mentioned, not its *Label*.

In the following example, we call the service *dhcp_scope_groupby* using TAGS and the statements *SELECT* and *GROUPBY* to count the scopes returned based on the value of the class parameter *information*.

```
https://192.168.0.1/rest/dhcp_scope_groupby?TAGS=dhcpscope.information&SELECT=count(*)%2Ctag_dhcpscope_information&GROUPBY=tag_dhcpscope_information
```

```
// In readable format the clause SELECT contains: count(*), tag_dhcpscope_information
```

Aggregating and Ordering Results

To aggregate and order the service result based on one or several class parameters, you must tag the class parameter(s) first, them in the statements *SELECT* and *GROUPBY* and finally indicate in the clause *ORDERBY* which parameter(s) order the result as follows.

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.<param>&SELECT=tag_<object-type>_<param>&GROUPBY=tag_<object-type>_<param>&ORDERBY=tag_<object-type>_<param> <order>
```

Once URL encoded, the call format is the following:

```
https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.<param>&SELECT=tag_<object-type>_<param>&GROUPBY=tag_<object-type>_<param>&ORDERBY=tag_<object-type>_<param>%20<order>
```

https://<your-SOLIDserver>/rest/<service-name>?TAGS=<object-type>.<param>

For more details, refer to the section [Tagging a Class Parameter](#).

&SELECT=tag_<object-type>_<param>

The statement that indicates which parameter is returned in output. For more details regarding *SELECT*, refer to the section [Aggregating Results based on Class Parameters](#) above.

&GROUPBY=tag_<object-type>_<param>

The statement that aggregates the results using the specified output parameter. For more details regarding *GROUPBY*, refer to the section [Aggregating Results based on Class Parameters](#) above.

&ORDERBY=tag_<object-type>_<param> <order>

The clause that indicates that the output parameters must be sorted, and in which order, it can be either **ASC** (ascending) or **DESC** (descending) and preceded by an encoded space: *tag_<object-type>_<param>%20<order>*.

The **<order>** is optional, if you do not specify any, the service returns information in ascending order. For more details regarding *ORDERBY*, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

In the following example, we call the service *dhcp_scope_groupby* using TAGS, the statements *SELECT* and *GROUPBY* to count the scopes returned and the clause *ORDERBY* to order the results all based on the value of the class parameter *information*.

```
https://192.168.0.1/rest/dhcp_scope_groupby?TAGS=dhcpscope.information&SELECT=count(*)%2C
```

```
tag_dhcpscope_information&GROUPBY=tag_dhcpscope_information&ORDERBY=tag_dhcpscope_information
```

```
// In readable format the clause SELECT contains: count(*),tag_dhcpscope_information
```

Expected Object Types in the TAGS

TAGS uses the database table names to identify the type of each object. The table below details the expected <object-type> of all the objects supporting TAGS described in this guide.

Table 4.1. Expected object types in the TAGS

| Module | Object | Expected type after ?TAGS= |
|----------------|--------------------|----------------------------|
| IPAM | Space | site |
| | Network (v4) | network |
| | Network (v6) | network6 |
| | Pool (v4) | pool |
| | Pool (v6) | pool6 |
| | IP address (v4) | ip |
| | IP address (v6) | ip6 |
| DHCP | Server (v4) | dhcp |
| | Server (v6) | dhcp6 |
| | Scope (v4) | dhcpscope |
| | Scope (v6) | dhcpscope6 |
| | Group (v4) | dhcpgroup |
| | Group (v6) | dhcpgroup6 |
| | Range (v4) | dhcprange |
| | Range (v6) | dhcprange6 |
| | Static (v4) | dhcpstatic |
| | Static (v6) | dhcpstatic6 |
| DNS | Server | dns |
| | Zone | dnszone |
| | View | dnsview |
| NOM | Folder | nomfolder |
| | Network object | nomnetobj |
| NetChange | Network device | iplnetdev |
| | Port | iplport |
| Workflow | Request | request |
| Device Manager | Device | hostdev |
| | Ports & interfaces | hostiface |
| VLAN Manager | Domain | vlmdomain |
| | Range | vlmrange |
| VRF | VRF | vrfobject |
| Administration | Group of users | grp |
| | User | usr |

Part II. IPAM Services

Table of Contents

| | |
|--|-----|
| 5. Space | 24 |
| ip_site_add | 25 |
| ip_site_list | 28 |
| ip_site_info | 31 |
| ip_site_count | 34 |
| group_site_add | 35 |
| group_site_delete | 37 |
| ip_site_delete | 39 |
| 6. IPv4 Network | 41 |
| ip_subnet_add | 42 |
| ip_block_subnet_list | 49 |
| ip_block_subnet_info | 56 |
| ip_block_subnet_count | 62 |
| ip_find_free_subnet | 63 |
| ip_block_subnet_groupby | 66 |
| ip_block_subnet_groupby_count | 68 |
| group_subnet_add | 70 |
| group_subnet_delete | 74 |
| ip_subnet_delete | 78 |
| 7. IPv6 Network | 82 |
| ip6_subnet6_add | 83 |
| ip6_block6_subnet6_list | 89 |
| ip6_block6_subnet6_info | 96 |
| ip6_block6_subnet6_count | 102 |
| ip6_find_free_subnet6 | 103 |
| ip6_block6_subnet6_groupby | 105 |
| ip6_block6_subnet6_groupby_count | 107 |
| group_subnet6_add | 109 |
| group_subnet6_delete | 112 |
| ip6_subnet6_delete | 115 |
| 8. IPv4 Pool | 118 |
| ip_pool_add | 119 |
| ip_pool_list | 123 |
| ip_pool_info | 128 |
| ip_pool_count | 132 |
| group_pool_add | 133 |
| group_pool_delete | 136 |
| ip_pool_delete | 139 |
| 9. IPv6 Pool | 141 |
| ip6_pool6_add | 142 |
| ip6_pool6_list | 146 |
| ip6_pool6_info | 151 |
| ip6_pool6_count | 155 |
| group_pool6_add | 156 |
| group_pool6_delete | 159 |
| ip6_pool6_delete | 161 |
| 10. IPv4 Address | 163 |
| ip_add | 164 |
| ip_address_list | 169 |
| ip_address_info | 176 |
| ip_address_count | 183 |

| | |
|----------------------------------|-----|
| ip_find_free_address | 184 |
| ip_free_address_list | 187 |
| ip_used_address_list | 194 |
| ip_address_groupby | 201 |
| ip_address_groupby_count | 203 |
| ip_delete | 205 |
| 11. IPv6 Address | 207 |
| ip6_address6_add | 208 |
| ip6_address6_list | 212 |
| ip6_address6_info | 218 |
| ip6_address6_count | 224 |
| ip6_find_free_address6 | 225 |
| ip6_address6_groupby | 228 |
| ip6_address6_groupby_count | 230 |
| ip6_address6_delete | 232 |
| 12. IPv4 Address Alias | 234 |
| ip_alias_add | 235 |
| ip_alias_list | 238 |
| ip_alias_count | 241 |
| ip_alias_delete | 242 |
| 13. IPv6 Address Alias | 244 |
| ip6_alias_add | 245 |
| ip6_alias_list | 248 |
| ip6_alias_count | 251 |
| ip6_alias_delete | 252 |

Chapter 5. Space

Name

ip_site_add — Add/Edit a space

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** site_name
- **Editing:** (site_id || site_name)

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space, each space must have a unique name.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_description

The description of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | | Can be edited | Yes |

parent_site_id

The database identifier (ID) of an existing space you want to set as the VLSM parent of the space you are adding/editing. This sets up a space-based VLSM organization.

| Type | Integer | Maximum length | N/A |
|---------------|---------|----------------|-----|
| Default value | 0 | Can be edited | Yes |

parent_site_name

The name of an existing space you want to set as the VLSM parent of the space you are adding/editing. This sets up a space-based VLSM organization.

| Type | String | Maximum length | N/A |
|------|--------|----------------|-----|
| | | | |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

site_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

site_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

site_is_template

The template status of the space you are adding/editing. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template. You can only set this parameter once, you cannot edit its value.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | No |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>_class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... . If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

ip_site_list — List the spaces

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

site_is_template

The template status of the space. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

site_id

The database identifier (ID) of the space.

tree_level

The database level of the space. The highest level is 0. If you set up a VLSM organization, it returns values between 0 and *n*.

tree_path

The path toward the space in the database as follows: <space-name>#. If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#... .

tree_id_path

The path toward the space in the database as follows: <space-ID>#. If you set up a VLSM organization, the path looks as follows: <highest-level-space-ID>#<child-space-ID>#<child-space-ID>#... .

site_name

The name of the space.

site_description

The description of the space.

parent_site_id

The database identifier (ID) of the VLSM parent space. 0 indicates that space has no parent space.

parent_site_name

The name of the VLSM parent space. # indicates that space has no parent space.

site_class_name

The name of the class applied to the space, it can be preceded by the class directory.

parent_site_class_name

The name of the class applied to the VLSM parent space, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
 - 1 indicates the object is enabled and managed.
 - 2 indicates the object is unmanaged, disabled or both depending on the context.
- By default, *row_enabled* is set to 1 when an object is created.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 5.1. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

site_class_parameters

The class parameters applied to the space and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

site_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

parent_site_class_parameters

The class parameters applied to the VLSM parent space and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

parent_site_class_parameters_properties

The inheritance and/or propagation properties of the class parameters returned in the parameter **parent_site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

ip_site_info — Display the properties of a space

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

site_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

Output Parameters

site_is_template

The template status of the space. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

site_id

The database identifier (ID) of the space.

tree_level

The database level of the space. The highest level is 0. If you set up a VLSM organization, it returns values between 0 and n.

tree_path

The path toward the space in the database as follows: <space-name>#. If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#... .

tree_id_path

The path toward the space in the database as follows: <space-ID>#. If you set up a VLSM organization, the path looks as follows: <highest-level-space-ID>#<child-space-ID>#<child-space-ID>#... .

site_name

The name of the space.

site_description

The description of the space.

parent_site_id

The database identifier (ID) of the VLSM parent space. 0 indicates that space has no parent space.

parent_site_name

The name of the VLSM parent space. # indicates that space has no parent space.

site_class_name

The name of the class applied to the space, it can be preceded by the class directory.

parent_site_class_name

The name of the class applied to the VLSM parent space, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 5.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

site_class_parameters

The class parameters applied to the space and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....

site_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

parent_site_class_parameters

The class parameters applied to the VLSM parent space and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

parent_site_class_parameters_properties

The inheritance and/or propagation properties of the class parameters returned in the parameter **parent_site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

ip_site_count — Count the number of spaces

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

group_site_add — Add a space to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (site_id || site_name))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
|---------------|--|----------------|-----|
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

group_site_delete — Remove a space from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (site_id || site_name))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

ip_site_delete — Delete a space

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(site_id || site_name)

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 6. IPv4 Network

Name

ip_subnet_add — Add/Edit an IPv4 block/subnet-type network

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is created.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

Note that to add a block-type network, setting the input parameter `subnet_level` to 0 is mandatory.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (`subnet_addr` && (`subnet_end_addr` || `subnet_size` || `subnet_mask` || `subnet_prefix`) && (`site_id` || `site_name` || `parent_subnet_id`))
- **Editing:** (`subnet_id` || (`subnet_addr` && (`subnet_end_addr` || `subnet_size` || `subnet_mask` || `subnet_prefix`) && (`site_id` || `site_name` || `parent_subnet_id`)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlsm_site_id

The database identifier (ID) of a VLSM child space of the space specified in `site_id`. If you specify an ID, the subnet-type network you are adding/editing is duplicated as a VLSM block-type network in the child space, with the same name but a different ID. This parameter serves the same purpose as `vlsm_site_name`.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlsm_site_name

The name of a VLSM child space of the space specified in `site_id`. If you specify a name, the subnet-type network you are adding/editing is duplicated as a VLSM block-type network in the child space, with the same name but a different ID. This parameter serves the same purpose as `vlsm_site_id`.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

subnet_id

The database identifier (ID) of the IPv4 network, a unique numeric key value automatically incremented when you add an IPv4 network. Use the ID to specify the IPv4 network of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_name

The name of the IPv4 network, each IPv4 network must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

subnet_addr

The start IP address of the IPv4 network, its first IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet_end_addr

The end IP address of the IPv4 network, its last IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet_size

The size of the IPv4 network, the number of IP addresses it contains.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_mask

The netmask of the IPv4 network. It is expressed in dot-decimal notation and defines the number of addresses the network contains.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_prefix

The prefix of the IPv4 network, an integer that defines the number of addresses the network contains.

| | | | |
|---------------|--|----------------|-----|
| Type | IPv4 prefix (integer between 1 and 32) | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_level

The level of the network within the space:

- Set it to *0* for a block-type network.
 - Set it to a value between *1* and *n* for a subnet-type network.
- If you set a value between *2* and *n*, you are setting a network-based VLSM organization where non terminal subnet-type networks can contain other subnet-type networks.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

parent_subnet_id

The database identifier (ID) of an existing IPv4 network you want to set as the parent of the IPv4 network you are adding/editing. You can specify a subnet-type network to set up a network-based VLSM organization.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

allow_tree_reparenting

A way to allow (1) or prevent (0) changing the parent of the network you are adding. Upon editing of the network, this parameter decides if you can associate it with a different parent network.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

relative_position

The position of the network within the hierarchy of networks of a VLSM organization. It calculates between *0* and *n* all the levels of the organization, its behavior depends on the value of the parameter **use_reversed_relative_position**:

- *use_reversed_relative_position=0* where *0* indicates a block-type network at the highest level possible, in a space-based organization, it belongs to the top space. The levels increment from *0* down to *n*, the lowest level you set up, within networks or spaces.
- *use_reversed_relative_position=1* where *1* indicates a network located at the lowest level of the organization, within networks or spaces. The levels increment from *0* up to *n*, the network at the highest level of the organization.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

use_reversed_relative_position

A way to determine if the calculation of the parameter **relative_position** should start from the top (*0*) or the bottom (*1*) of the VLSM organization.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

subnet_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
|------|--------|----------------|-----|

| | | | |
|----------------------|--|----------------------|-----|
| Default value | | Can be edited | Yes |
|----------------------|--|----------------------|-----|

network_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

subnet_class_parameters

Deprecated, replaced by **network_class_parameters**.

subnet_class_parameters_properties

Deprecated, replaced by **network_class_parameters_properties**.

permit_invalid

A way to authorize (1) IPv4 networks overlapping within a space.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

permit_overlap

Deprecated, replaced by **permit_invalid**.

permit_no_block

A way to force the creation of an IPv4 subnet-type network. If set to 1, you can create a subnet-type network even if no block-type network matching the start address exists.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

allow_block_creation

Internal use. Not documented.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

is_terminal

A way to determine if a network can contain other networks. If set to 1, the network is terminal and cannot contain other subnet-type networks. By essence, block-type networks are non-terminal and are always set to 0.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmvlan_id

The database identifier (ID) of the VLAN you want to associate with the network.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

enabled

Deprecated, replaced by **row_enabled**.

row_enabled

The object activation status.

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.
- If set to 2, the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: 1 2 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

network_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.

- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Example

In the example below, we call the service **ip_subnet_add** with Python (Requests) to add a block-type network.

Example 6.1. Calling the service ip_subnet_add using Python

```
import requests  
  
url = "https://solid.intranet/rest/ip_subnet_add"
```

```
querystring =  
{"subnet_addr": "192.168.0.0", "subnet_level": "0", "subnet_prefix": "24", "site_id": "2", "subnet_name": "internal-network"}  
  
headers = {  
    'x-ipm-username': "aXBtYWRtaW4=",  
    'x-ipm-password': "YWRtaW4=",  
    'cache-control': "no-cache"  
}  
  
response = requests.request("POST", url, headers=headers, params=querystring)  
  
print(response.text)
```

Name

ip_block_subnet_list — List the IPv4 block/subnet-type networks

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

type

The type of the network.

subnet_id

The database identifier (ID) of the IPv4 network.

start_ip_addr

The first IP address of the IPv4 network, in hexadecimal format.

start_hostaddr

The human readable version of the parameter **start_ip_addr**.

end_ip_addr

The last IP address of the IPv4 network, in hexadecimal format.

end_hostaddr

The human readable version of the parameter **end_ip_addr**.

subnet_name

The name of the IPv4 network. *Default* indicates that the network is an orphan network.

subnet_size

The number of IP addresses the IPv4 network contains.

vlsm_block_id

The database identifier (ID) of the IPv4 VLSM block-type network duplicated, in a VLSM child space, from the network. *0* indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlmvlan_id

The database identifier (ID) of the VLAN associated with the network.

subnet_level

The level of the network within the space. It returns values between *0* (block-type network) and *n* (subnet-type network). A value higher than *1* indicates a VLSM organization where a block-type network can belong to another subnet-type network.

subnet_path

The path toward the network in the database from the containing block-type network down to the subnet-type network: <*block-network-start-IP*>#<*block-network-ID*>#<*subnet-network-start-IP*>#<*subnet-network-ID*>. The IP address is returned in hexadecimal format.

- In network-based VLSM organizations, the path includes all the subnet-type networks there are from the containing block-type network down to the subnet-type network specified in *subnet_id*.

- In space-based VLSM organizations, the path includes the block-type network of the top parent space and all the subnet-type networks there are until the network specified in *subnet_id*. Only one block-type network is returned.

subnet_class_name

The name of the class applied to the IPv4 network, it can be preceded by the class directory.

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. 0 indicates that the network has no parent network.

vlsm_subnet_id

The database identifier (ID) of the IPv4 subnet-type network, located in the VLSM parent space, from which the network was duplicated. 0 indicates that the network is not a VLSM block-type network duplicated from a parent space.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

subnet_is_valid

The network validity. A valid network (1) has a size, prefix and/or netmask that match.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

waiting_state

The state of the exchange between SOLIDserver and the RIPE for the assigned network:

Table 6.1. *waiting_state* possible values

| Status | Description |
|--------------------|---|
| must_send_mail_add | An email must be sent to the RIPE to notify them of a subnet-type network creation. |
| wait_mail_add | A network creation email was sent to the RIPE, no reply has been received yet. |
| must_send_mail_del | An email must be sent to the RIPE to notify them of a subnet-type network deletion. |
| wait_mail_del | A network deletion email was sent to the RIPE, no reply has been received yet. |
| wait_aw_confirm | The number of IP addresses of the assigned network exceeds the Assignment Window declared during your RIPE configuration. |

waiting_status

The status of a RIPE assigned network within SOLIDserver until it is confirmed that you can create or delete it. If set to 1, it is about to be created. If set to 2, it is about to be deleted.

is_terminal

A way to determine if a network can contain other networks. If set to 1, the network is terminal and cannot contain other subnet-type networks. Block-type networks are always set to 0.

subnet_allocated_size

The sum of the size of all the subnet-type networks that belong to the block-type network.

subnet_allocated_percent

The percentage of subnet-type networks the non-terminal network contains.

subnet_used_size

The sum of the size of all the terminal networks within the block-type network. This sum includes the terminal networks that might belong to non-terminal subnet-type networks.

subnet_used_percent

The percentage of terminal networks the non-terminal network contains.

subnet_ip_used_size

The number of IP addresses *In use* in terminal networks.

subnet_ip_used_percent

The percentage of IP addresses *In use* in terminal networks.

subnet_ip_free_size

The total number of free addresses, for terminal networks only. It excludes the network and broadcast IP address.

is_in_orphan

A way to determine if the network has a parent (0) or if it belongs to a container *Orphan networks* (1).

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network.

site_description

The description of the space the object belongs to.

site_name

The name of the space the object belongs to.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

tree_level

The database level of the space the object belongs to. If you set up a VLSM organization, it returns values between between 0 (the highest level) and *n*.

tree_path

The database path toward the space the object belongs to as follows: <space-name>#. If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#... .

tree_id_path

The database path toward the space the object belongs to as follows: <space-ID>#. If you set up a VLSM organization, the path looks as follows: <highest-level-space-ID>#<child-space-ID>#<child-space-ID>#... .

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

parent_subnet_name

The name of the parent IPv4 network:

- # indicates that the network has no parent network.
- *Default* indicates that the network belongs to an orphan network.

parent_start_ip_addr

The first IP address of the parent IPv4 network, in hexadecimal format.

parent_end_ip_addr

The last IP address of the parent IPv4 network, in hexadecimal format.

parent_subnet_size

The number of IP addresses of the network parent.

parent_subnet_level

The level of the parent network within the space. It returns values between *0* (block-type network) and *n* (subnet-type network). A value higher than *1* indicates a VLSM organization where a block-type network can belong to another subnet-type network.

parent_subnet_path

The path toward the parent network in the database. # indicates the network has no parent network.

parent_subnet_class_name

The name of the class applied to the parent IPv4 network, it can be preceded by the class directory.

parent_is_terminal

A way to determine if the parent network is terminal (*1*) or non-terminal (*0*).

parent_vlsm_subnet_id

The database identifier (ID) of the IPv4 subnet-type network, located in the VLSM parent space, from which the parent network was duplicated. *0* indicates that the parent network is not a VLSM block-type network duplicated from a parent space.

parent_site_id

The database identifier (ID) of the space where is located the parent network. *0* indicates that the network has no parent network.

parent_site_name

The name of the space where is located the parent network. # indicates that the network has no parent network.

site_parent_site_id

The database identifier (ID) of the VLSM parent of the space where is located the network. *0* indicates that the space where is located the network has no parent space.

vlsm_site_id

The database identifier (ID) of the VLSM child space where the network is duplicated as a VLSM block-type network. *0* indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlsm_site_name

The name of the VLSM child space where the network is duplicated as a VLSM block-type network. *0* indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlmvlan_vlan_id

The VLAN identifier (ID) of the VLAN associated with the network.

vlmvlan_name

The name of the VLAN associated with the network.

vlmdomain_id

The database identifier (ID) of the VLAN domain associated with the network.

vlmdomain_name

The name of the VLAN domain associated with the network.

vlmrang_id

The database identifier (ID) of the VLAN range associated with the network.

vlmrange_name

The name of the VLAN range associated with the network.

trace_creation_date

The creation date of the IPv4 network, in decimal UNIX date format.

trace_last_update_date

The last time the IPv4 network was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv4 network.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv4 network.

trace_creation_origin

The name of the module where the IPv4 network addition originated.

trace_creation_exec_stack

The call stack of the IPv4 network operation details, as follows: <service1>&<service2>&<service3>....

trace_creation_usr_login

The login of the user who added the IPv4 network.

trace_creation_origin_usr_login

The login of the user who requested the IPv4 network.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 6.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

subnet_class_parameters

The class parameters applied to the IPv4 network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

subnet_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:

<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&.... .

parent_subnet_class_parameters

The class parameters applied to the parent IPv4 network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

parent_subnet_class_parameters_properties

The inheritance and/or propagation properties of the class parameters returned in the parameter **parent_subnet_class_parameter**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Example

In the example below, we call the service **ip_block_subnet_list** with PHP (cURL) using the clauses *WHERE* and *ORDERBY* and the parameter *limit* to list the ten first /24 networks in ascending order. For more details regarding the use of class parameters in the clause, refer to the chapter [Calling Services With TAGS](#).

Example 6.2. Calling the service ip_block_subnet_list using PHP, WHERE and ORDERBY

```
<?php
$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/ip_block_subnet_list?WHERE".
                    "=subnet_size%3D%27256%27&ORDERBY=start_ip_addr&limit=10",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));
$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Name

ip_block_subnet_info — Display the properties of an IPv4 block/subnet-type network

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

subnet_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

subnet_id

The database identifier (ID) of the IPv4 network, a unique numeric key value automatically incremented when you add an IPv4 network. Use the ID to specify the IPv4 network of your choice.

Output Parameters

type

The type of the network.

subnet_id

The database identifier (ID) of the IPv4 network.

start_ip_addr

The first IP address of the IPv4 network, in hexadecimal format.

start_hostaddr

The human readable version of the parameter `start_ip_addr`.

end_ip_addr

The last IP address of the IPv4 network, in hexadecimal format.

end_hostaddr

The human readable version of the parameter **end_ip_addr**.

subnet_name

The name of the IPv4 network. *Default* indicates that the network is an orphan network.

subnet_size

The number of IP addresses the IPv4 network contains.

vlsm_block_id

The database identifier (ID) of the IPv4 VLSM block-type network duplicated, in a VLSM child space, from the network. *0* indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlmvlan_id

The database identifier (ID) of the VLAN associated with the network.

subnet_level

The level of the network within the space. It returns values between *0* (block-type network) and *n* (subnet-type network). A value higher than *1* indicates a VLSM organization where a block-type network can belong to another subnet-type network.

subnet_path

The path toward the network in the database from the containing block-type network down to the subnet-type network: <block-network-start-IP>#<block-network-ID>#<subnet-network-start-IP>#<subnet-network-ID>. The IP address is returned in hexadecimal format.

- In network-based VLSM organizations, the path includes all the subnet-type networks there are from the containing block-type network down to the subnet-type network specified in *subnet_id*.
- In space-based VLSM organizations, the path includes the block-type network of the top parent space and all the subnet-type networks there are until the network specified in *subnet_id*. Only one block-type network is returned.

subnet_class_name

The name of the class applied to the IPv4 network, it can be preceded by the class directory.

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. *0* indicates that the network has no parent network.

vlsm_subnet_id

The database identifier (ID) of the IPv4 subnet-type network, located in the VLSM parent space, from which the network was duplicated. *0* indicates that the network is not a VLSM block-type network duplicated from a parent space.

row_enabled

The object activation status:

- *0* indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- *1* indicates the object is enabled and managed.
- *2* indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to *1* when an object is created.

subnet_is_valid

The network validity. A valid network (*1*) has a size, prefix and/or netmask that match.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

waiting_state

The state of the exchange between SOLIDserver and the RIPE for the assigned network:

Table 6.3. *waiting_state* possible values

| Status | Description |
|--------------------|---|
| must_send_mail_add | An email must be sent to the RIPE to notify them of a subnet-type network creation. |
| wait_mail_add | A network creation email was sent to the RIPE, no reply has been received yet. |
| must_send_mail_del | An email must be sent to the RIPE to notify them of a subnet-type network deletion. |
| wait_mail_del | A network deletion email was sent to the RIPE, no reply has been received yet. |
| wait_aw_confirm | The number of IP addresses of the assigned network exceeds the Assignment Window declared during your RIPE configuration. |

waiting_status

The status of a RIPE assigned network within SOLIDserver until it is confirmed that you can create or delete it. If set to 1, it is about to be created. If set to 2, it is about to be deleted.

is_terminal

A way to determine if a network can contain other networks. If set to 1, the network is terminal and cannot contain other subnet-type networks. Block-type networks are always set to 0.

subnet_allocated_size

The sum of the size of all the subnet-type networks that belong to the block-type network.

subnet_allocated_percent

The percentage of subnet-type networks the non-terminal network contains.

subnet_used_size

The sum of the size of all the terminal networks within the block-type network. This sum includes the terminal networks that might belong to non-terminal subnet-type networks.

subnet_used_percent

The percentage of terminal networks the non-terminal network contains.

subnet_ip_used_size

The number of IP addresses *In use* in terminal networks.

subnet_ip_used_percent

The percentage of IP addresses *In use* in terminal networks.

subnet_ip_free_size

The total number of free addresses, for terminal networks only. It excludes the network and broadcast IP address.

is_in_orphan

A way to determine if the network has a parent (0) or if it belongs to a container *Orphan networks* (1).

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network.

site_description

The description of the space the object belongs to.

site_name

The name of the space the object belongs to.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

tree_level

The database level of the space the object belongs to. If you set up a VLSM organization, it returns values between 0 (the highest level) and n .

tree_path

The database path toward the space the object belongs to as follows: <space-name># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#....

tree_id_path

The database path toward the space the object belongs to as follows: <space-ID># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-ID>#<child-space-ID>#<child-space-ID>#....

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

parent_subnet_name

The name of the parent IPv4 network:

- # indicates that the network has no parent network.
- Default indicates that the network belongs to an orphan network.

parent_start_ip_addr

The first IP address of the parent IPv4 network, in hexadecimal format.

parent_end_ip_addr

The last IP address of the parent IPv4 network, in hexadecimal format.

parent_subnet_size

The number of IP addresses of the network parent.

parent_subnet_level

The level of the parent network within the space. It returns values between 0 (block-type network) and n (subnet-type network). A value higher than 1 indicates a VLSM organization where a block-type network can belong to another subnet-type network.

parent_subnet_path

The path toward the parent network in the database. # indicates the network has no parent network.

parent_subnet_class_name

The name of the class applied to the parent IPv4 network, it can be preceded by the class directory.

parent_is_terminal

A way to determine if the parent network is terminal (1) or non-terminal (0).

parent_vlsm_subnet_id

The database identifier (ID) of the IPv4 subnet-type network, located in the VLSM parent space, from which the parent network was duplicated. 0 indicates that the parent network is not a VLSM block-type network duplicated from a parent space.

parent_site_id

The database identifier (ID) of the space where is located the parent network. 0 indicates that the network has no parent network.

parent_site_name

The name of the space where is located the parent network. # indicates that the network has no parent network.

site_parent_site_id

The database identifier (ID) of the VLSM parent of the space where is located the network.
0 indicates that the space where is located the network has no parent space.

vlsm_site_id

The database identifier (ID) of the VLSM child space where the network is duplicated as a VLSM block-type network. 0 indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlsm_site_name

The name of the VLSM child space where the network is duplicated as a VLSM block-type network. 0 indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlmvlan_vlan_id

The VLAN identifier (ID) of the VLAN associated with the network.

vlmvlan_name

The name of the VLAN associated with the network.

vlmdomain_id

The database identifier (ID) of the VLAN domain associated with the network.

vlmdomain_name

The name of the VLAN domain associated with the network.

vlmrang_id

The database identifier (ID) of the VLAN range associated with the network.

vlmrang_name

The name of the VLAN range associated with the network.

trace_creation_date

The creation date of the IPv4 network, in decimal UNIX date format.

trace_last_update_date

The last time the IPv4 network was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv4 network.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv4 network.

trace_creation_origin

The name of the module where the IPv4 network addition originated.

trace_creation_exec_stack

The call stack of the IPv4 network operation details, as follows: <service1>&<service2>&<service3>... .

trace_creation_usr_login

The login of the user who added the IPv4 network.

trace_creation_origin_usr_login

The login of the user who requested the IPv4 network.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 6.4. Multi-status severity levels

| Message number | Severity | Description |
|-----------------------|-----------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

subnet_class_parameters

The class parameters applied to the IPv4 network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

subnet_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&.... .

parent_subnet_class_parameters

The class parameters applied to the parent IPv4 network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

parent_subnet_class_parameters_properties

The inheritance and/or propagation properties of the class parameters returned in the parameter **parent_subnet_class_parameter**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

ip_block_subnet_count — Count the number of IPv4 block/subnet-type networks

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

ip_find_free_subnet — List the free IPv4 subnet-type networks

Description

This service allows you to list the 10 first free IPv4 subnet-type networks, terminal or non terminal.

You must execute the service using **rpc**.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(prefix || size)

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

prefix

The prefix of the subnet-type network(s) you are looking for, an integer that defines the number of addresses the network contains.

| Type | IPv4 prefix (integer between 1 and 32) | Maximum length | N/A |
|---------------|--|----------------|-----|
| Default value | N/A | Can be edited | Yes |

size

The size of the subnet-type network(s) you are looking, an integer that indicates the number of IP addresses they contain.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

max_find

The maximum number of IPv4 networks to be returned by the service. You can use it to return more than 10 results.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

begin_addr

The first IPv4 address of the range of addresses where you are looking for free networks.

| Type | IPv4 address | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

end_addr

The last IPv4 address of the range of addresses where you are looking for free networks.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

| | | | |
|---------------|-----|----------------|-----|
| Type | | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

block_id

The database identifier (ID) of an existing non-terminal IPv4 network. Use the ID to specify the IPv4 network of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

use_searched_path

A way to filter the search for subnet-type networks based on the specified **block_id**. If set to 0, the service returns free subnet-type networks within the specified block-type network. If set to 1, the service returns free subnet-type networks within the specified block-type network and within all the non-terminal subnet-type networks it might contain.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

Output Parameters

start_ip_addr

The first IP address of the IPv4 network, in hexadecimal format.

start_hostaddr

The human readable version of the parameter **start_ip_addr**.

block_name

The name of the non-terminal IPv4 network the free subnet-type network(s) belongs to.

cost

An integer between 0 and n that evaluates the best range of IP addresses within a block-type network to create a subnet-type network and avoid fragmentation. The lower the cost, the better the position is. The lowest costs are always returned first.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause **WHERE**.

block_id

The database identifier (ID) of the non-terminal IPv4 network the free subnet-type network(s) belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

Example

In the example below, we call the service **ip_find_free_subnet** with Ruby (NET::Http) to look for /30 networks in a specific space.

Example 6.3. Calling the service ip_find_free_subnet using Ruby

```
require 'uri'
require 'net/http'

url = URI("https://solid.intranet/rpc/ip_find_free_subnet?prefix=30&site_id=2")

http = Net::HTTP.new(url.host, url.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE

request = Net::HTTP::Options.new(url)
request["x-ipm-username"] = 'aXBtYWRtaW4='
request["x-ipm-password"] = 'YWRtaW4='
request["cache-control"] = 'no-cache'

response = http.request(request)
puts response.read_body
```

Name

ip_block_subnet_groupby — Group IPv4 block/subnet-type networks by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

ip_block_subnet_groupby_count — Count the number of IPv4 block/subnet-type networks grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

group_subnet_add — Add an IPv4 block/subnet-type network to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (subnet_id || (subnet_addr && (subnet_end_addr || subnet_size || subnet_mask || subnet_prefix) && (site_id || site_name || parent_subnet_id))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

subnet_id

The database identifier (ID) of the IPv4 network, a unique numeric key value automatically incremented when you add an IPv4 network. Use the ID to specify the IPv4 network of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. Use the ID to specify the parent IPv4 network of your choice.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_name

The name of the IPv4 network.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

subnet_addr

The start IP address of the IPv4 network, its first IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet_end_addr

The end IP address of the IPv4 network, its last IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_size

The size of the IPv4 network, the number of IP addresses it contains.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_mask

The netmask of the IPv4 network. It is expressed in dot-decimal notation and defines the number of addresses the network contains.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_prefix

The prefix of the IPv4 network, an integer that defines the number of addresses the network contains.

| | | | |
|----------------------|--|-----------------------|-----|
| Type | IPv4 prefix (integer between 1 and 32) | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_level

The level of the network within the space:

- Set it to 0 for a block-type network.
- Set it to a value between 1 and n for a subnet-type network.

If you set a value between 2 and n , you are setting a network-based VLSM organization where non terminal subnet-type networks can contain other subnet-type networks.

| | | | |
|---------------|------------------|----------------|-----|
| Type | Integer ≥ 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

relative_position

The position of the network within the hierarchy of networks of a VLSM organization. It calculates between 0 and n all the levels of the organization, its behavior depends on the value of the parameter **use_reversed_relative_position**:

- *use_reversed_relative_position=0* where 0 indicates a block-type network at the highest level possible, in a space-based organization, it belongs to the top space. The levels increment from 0 down to n , the lowest level you set up, within networks or spaces.
- *use_reversed_relative_position=1* where 1 indicates a network located at the lowest level of the organization, within networks or spaces. The levels increment from 0 up to n , the network at the highest level of the organization.

| | | | |
|---------------|------------------|----------------|-----|
| Type | Integer ≥ 0 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

use_reversed_relative_position

A way to determine if the calculation of the parameter **relative_position** should start from the top (0) or the bottom (1) of the VLSM organization.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0 . If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Example

In the example below, we call the service **group_subnet_add** with Python (Requests) to add a block-type network in a group of users.

Example 6.4. Calling the service group_subnet_add using Python

```
import requests

url = "https://solid.intranet/rest/group_subnet_add"

querystring =
{"grp_name": "regular", "subnet_id": "241", "relative_position": "0", "site_id": "44"}

headers = {
    'x-ipm-username': "aXBtYWRtaW4=",
    'x-ipm-password': "YWRtaW4=",
    'cache-control': "no-cache"
}

response = requests.request("POST", url, headers=headers, params=querystring)

print(response.text)
```

Name

group_subnet_delete — Remove an IPv4 block/subnet-type network from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (subnet_id || (subnet_addr && (subnet_end_addr || subnet_size || subnet_mask || subnet_prefix) && (site_id || site_name || parent_subnet_id))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet_id

The database identifier (ID) of the IPv4 network, a unique numeric key value automatically incremented when you add an IPv4 network. Use the ID to specify the IPv4 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|------|-------------|----------------|-----|
| | | | |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. Use the ID to specify the parent IPv4 network of your choice.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_name

The name of the IPv4 network.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

subnet_addr

The start IP address of the IPv4 network, its first IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet_end_addr

The end IP address of the IPv4 network, its last IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_size

The size of the IPv4 network, the number of IP addresses it contains.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_mask

The netmask of the IPv4 network. It is expressed in dot-decimal notation and defines the number of addresses the network contains.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_prefix

The prefix of the IPv4 network, an integer that defines the number of addresses the network contains.

| | | | |
|----------------------|--|-----------------------|-----|
| Type | IPv4 prefix (integer between 1 and 32) | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_level

The level of the network within the space:

- Set it to 0 for a block-type network.
- Set it to a value between 1 and n for a subnet-type network.

If you set a value between 2 and n , you are setting a network-based VLSM organization where non terminal subnet-type networks can contain other subnet-type networks.

| | | | |
|---------------|------------------|----------------|-----|
| Type | Integer ≥ 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

relative_position

The position of the network within the hierarchy of networks of a VLSM organization. It calculates between 0 and n all the levels of the organization, its behavior depends on the value of the parameter **use_reversed_relative_position**:

- *use_reversed_relative_position=0* where 0 indicates a block-type network at the highest level possible, in a space-based organization, it belongs to the top space. The levels increment from 0 down to n , the lowest level you set up, within networks or spaces.
- *use_reversed_relative_position=1* where 1 indicates a network located at the lowest level of the organization, within networks or spaces. The levels increment from 0 up to n , the network at the highest level of the organization.

| | | | |
|---------------|------------------|----------------|-----|
| Type | Integer ≥ 0 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

use_reversed_relative_position

A way to determine if the calculation of the parameter **relative_position** should start from the top (0) or the bottom (1) of the VLSM organization.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0 . If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

ip_subnet_delete — Delete an IPv4 block/subnet-type network

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(subnet_id || (subnet_addr && (subnet_end_addr || subnet_size || subnet_mask || subnet_prefix) && (site_id || site_name || parent_subnet_id)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet_id

The database identifier (ID) of the IPv4 network, a unique numeric key value automatically incremented when you add an IPv4 network. Use the ID to specify the IPv4 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. Use the ID to specify the parent IPv4 network of your choice.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet_name

The name of the IPv4 network.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet_addr

The start IP address of the IPv4 network, its first IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet_end_addr

The end IP address of the IPv4 network, its last IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_size

The size of the IPv4 network, the number of IP addresses it contains.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_mask

The netmask of the IPv4 network. It is expressed in dot-decimal notation and defines the number of addresses the network contains.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_prefix

The prefix of the IPv4 network, an integer that defines the number of addresses the network contains.

| | | | |
|---------------|--|----------------|-----|
| Type | IPv4 prefix (integer between 1 and 32) | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_level

The level of the network within the space:

- Set it to *0* for a block-type network.
- Set it to a value between *1* and *n* for a subnet-type network.

If you set a value between *2* and *n*, you are setting a network-based VLSM organization where non terminal subnet-type networks can contain other subnet-type networks.

| | | | |
|---------------|------------------|----------------|-----|
| Type | Integer ≥ 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

relative_position

The position of the network within the hierarchy of networks of a VLSM organization. It calculates between *0* and *n* all the levels of the organization, its behavior depends on the value of the parameter **use_reversed_relative_position**:

- *use_reversed_relative_position=0* where *0* indicates a block-type network at the highest level possible, in a space-based organization, it belongs to the top space. The levels increment from *0* down to *n*, the lowest level you set up, within networks or spaces.

- *use_reversed_relative_position=1* where 1 indicates a network located at the lowest level of the organization, within networks or spaces. The levels increment from 0 up to *n*, the network at the highest level of the organization.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

use_reversed_relative_position

A way to determine if the calculation of the parameter **relative_position** should start from the top (0) or the bottom (1) of the VLSM organization.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Example

In the example below, we call the service **ip_subnet_delete** with PHP (cURL).

Example 6.5. Calling the service ip_subnet_delete using PHP

```
<?php
$curl = curl_init();
```

```
curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/ip_subnet_delete?".
        "subnet_id=241&relative_position=0&site_id=44",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "DELETE",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Chapter 7. IPv6 Network

Name

ip6_subnet6_add — Add/Edit an IPv6 block/subnet-type network

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (subnet6_addr && (subnet6_end_addr || subnet6_prefix) && (site_id || site_name || parent_subnet6_id))
- **Editing:** (subnet6_id || (subnet6_addr && (subnet6_end_addr || subnet6_prefix) && (site_id || site_name || parent_subnet6_id)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

vlsm_site_id

The database identifier (ID) of a VLSM child space of the space specified in *site_id*. If you specify an ID, the subnet-type network you are adding/editing is duplicated as a VLSM block-type network in the child space, with the same name but a different ID. This parameter serves the same purpose as *vlsm_site_name*.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlsm_site_name

The name of a VLSM child space of the space specified in *site_id*. If you specify a name, the subnet-type network you are adding/editing is duplicated as a VLSM block-type network in the child space, with the same name but a different ID. This parameter serves the same purpose as *vlsm_site_id*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

subnet6_id

The database identifier (ID) of the IPv6 network, a unique numeric key value automatically incremented when you add an IPv6 network. Use the ID to specify the IPv6 network of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet6_name

The name of the IPv6 network, each IPv6 network must have a unique name.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

subnet6_addr

The start IP address of the IPv6 network, its first IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet6_end_addr

The end IP address of the IPv6 network, its last IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet6_prefix

The prefix of the IPv6 network, an integer that defines the number of address the network contains.

| | | | |
|----------------------|---|-----------------------|-----|
| Type | IPv6 prefix (integer between 1 and 128) | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet_level

The level of the network within the space:

- Set it to *0* for a block-type network.
- Set it to a value between *1* and *n* for a subnet-type network.

If you set a value between *2* and *n*, you are setting a network-based VLSM organization where non terminal subnet-type networks can contain other subnet-type networks.

| | | | |
|----------------------|------------------|-----------------------|-----|
| Type | Integer ≥ 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

parent_subnet6_id

The database identifier (ID) of an existing IPv6 network you want to set as the parent of the IPv6 network you are adding/editing. You can specify a subnet-type network to set up a network-based VLSM organization.

| | | | |
|-------------|------------------|-----------------------|-----|
| Type | Integer ≥ 0 | Maximum length | N/A |
|-------------|------------------|-----------------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

allow_tree_reparenting

A way to allow (1) or prevent (0) changing the parent of the network you are adding. Upon editing of the network, this parameter decides if you can associate it with a different parent network.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

relative_position

The position of the network within the hierarchy of networks of a VLSM organization. It calculates between 0 and n all the levels of the organization, its behavior depends on the value of the parameter **use_reversed_relative_position**:

- **use_reversed_relative_position=0** where 0 indicates a block-type network at the highest level possible, in a space-based organization, it belongs to the top space. The levels increment from 0 down to n , the lowest level you set up, within networks or spaces.
- **use_reversed_relative_position=1** where 1 indicates a network located at the lowest level of the organization, within networks or spaces. The levels increment from 0 up to n , the network at the highest level of the organization.

| | | | |
|----------------------|------------------|-----------------------|-----|
| Type | Integer ≥ 0 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

use_reversed_relative_position

A way to determine if the calculation of the parameter **relative_position** should start from the top (0) or the bottom (1) of the VLSM organization.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

subnet6_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

network6_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

subnet6_class_parameters

Deprecated, replaced by **network6_class_parameters**.

subnet6_class_parameters_properties

Deprecated, replaced by **network6_class_parameters_properties**.

permit_invalid

A way to authorize (1) IPv6 networks overlapping within a space.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

permit_overlap

Deprecated, replaced by **permit_invalid**.

permit_no_block6

A way to force the creation of an IPv6 subnet-type network. If set to 1, you can create a subnet-type network even if no block-type network matching the start address exists.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

allow_block6_creation

Internal use. Not documented.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

is_terminal

A way to determine if a network can contain other networks. If set to 1, the network is terminal and cannot contain other subnet-type networks. By essence, block-type networks are non-terminal and are always set to 0.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmvlan_id

The database identifier (ID) of the VLAN you want to associate with the network.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

enabled

Deprecated, replaced by **row_enabled**.

row_enabled

The object activation status.

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.
- If set to 2, the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: 1 2 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

network6_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... . If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

ip6_block6_subnet6_list — List the IPv6 block/subnet-type networks

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

type

The type of the network.

subnet6_id

The database identifier (ID) of the IPv6 network.

start_ip6_addr

The first IP address of the IPv6 network, in hexadecimal format.

start_hostaddr

The human readable version of the parameter **start_ip6_addr**.

end_ip6_addr

The last IP address of the IPv6 network, in hexadecimal format.

end_hostaddr

The human readable version of the parameter **end_ip6_addr**.

subnet6_name

The name of the IPv6 network.

subnet_size

The number of IP addresses the IPv6 network contains.

vlsm_block6_id

The database identifier (ID) of the IPv6 VLSM block-type network duplicated, in a VLSM child space, from the network. 0 indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlmvlan_id

The database identifier (ID) of the VLAN associated with the network.

subnet_level

The level of the network within the space. It returns values between 0 (block-type network) and *n* (subnet-type network). A value higher than 1 indicates a VLSM organization where a block-type network can belong to another subnet-type network.

subnet_path

The path toward the network in the database from the containing block-type network down to the subnet-type network: <block-network-start-IP>#<block-network-ID>#<subnet-network-start-IP>#<subnet-network-ID>. The IP address is returned in hexadecimal format.

- In network-based VLSM organizations, the path includes all the subnet-type networks there are from the containing block-type network down to the subnet-type network specified in *subnet_id*.

- In space-based VLSM organizations, the path includes the block-type network of the top parent space and all the subnet-type networks there are until the network specified in *subnet_id*. Only one block-type network is returned.

subnet6_class_name

The name of the class applied to the IPv6 network, it can be preceded by the class directory.

parent_subnet6_id

The database identifier (ID) of the parent IPv6 network. 0 indicates that the network has no parent network.

vlsm_subnet6_id

The database identifier (ID) of the IPv6 subnet-type network, located in the VLSM parent space, from which the network was duplicated. 0 indicates that the network is not a VLSM block-type network duplicated from a parent space.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

subnet6_is_valid

The network validity. A valid network (1) has a prefix and last IP address that match.

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

waiting_state

The state of the exchange between SOLIDserver and the RIPE for the assigned network:

Table 7.1. *waiting_state* possible values

| Status | Description |
|--------------------|---|
| must_send_mail_add | An email must be sent to the RIPE to notify them of a subnet-type network creation. |
| wait_mail_add | A network creation email was sent to the RIPE, no reply has been received yet. |
| must_send_mail_del | An email must be sent to the RIPE to notify them of a subnet-type network deletion. |
| wait_mail_del | A network deletion email was sent to the RIPE, no reply has been received yet. |
| wait_aw_confirm | The number of IP addresses of the assigned network exceeds the Assignment Window declared during your RIPE configuration. |

waiting_status

The status of a RIPE assigned network within SOLIDserver until it is confirmed that you can create or delete it. If set to 1, it is about to be created. If set to 2, it is about to be deleted.

is_terminal

A way to determine if a network can contain other networks. If set to 1, the network is terminal and cannot contain other subnet-type networks. Block-type networks are always set to 0.

subnet6_prefix

The prefix of the IPv6 network.

percent_allocated

The percentage of subnet-type networks the non-terminal network contains.

percent_used

The percentage of terminal networks the non-terminal network contains.

is_in_orphan

A way to determine if the network has a parent (0) or if it belongs to a container *Orphan networks* (1).

site_description

The description of the space the object belongs to.

site_name

The name of the space the object belongs to.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

tree_level

The database level of the space the object belongs to. If you set up a VLSM organization, it returns values between between 0 (the highest level) and *n*.

tree_path

The database path toward the space the object belongs to as follows: <space-name># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#... .

tree_id_path

The database path toward the space the object belongs to as follows: <space-ID># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-ID>#<child-space-ID>#<child-space-ID>#... .

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

parent_subnet6_name

The name of the parent IPv6 network. # indicates that the network has no parent network.

parent_start_ip6_addr

The first IP address of the parent IPv6 network, in hexadecimal format.

parent_end_ip6_addr

The last IP address of the parent IPv6 network, in hexadecimal format.

parent_subnet_size

The number of IP addresses of the network parent, in hexadecimal format.

parent_subnet_level

The level of the parent network within the space. It returns values between 0 (block-type network) and *n* (subnet-type network). A value higher than 1 indicates a VLSM organization where a block-type network can belong to another subnet-type network.

parent_subnet_path

The path toward the parent network in the database. # indicates the network has no parent network.

parent_subnet6_class_name

The name of the class applied to the parent IPv6 network, it can be preceded by the class directory.

parent_is_terminal

A way to determine if the parent network is terminal (1) or non-terminal (0).

parent_vlsm_subnet6_id

The database identifier (ID) of the IPv6 subnet-type network, located in the VLSM parent space, from which the parent network was duplicated. 0 indicates that the parent network is not a VLSM block-type network duplicated from a parent space.

parent_subnet6_prefix

The prefix of the parent of the IPv6 network the object belongs to.

parent_percent_allocated

The percentage of subnet-type networks the parent network contains.

parent_percent_used

The percentage of terminal networks the parent network contains.

parent_site_id

The database identifier (ID) of the space where is located the parent network. 0 indicates that the network has no parent network.

parent_site_name

The name of the space where is located the parent network. # indicates that the network has no parent network.

site_parent_site_id

The database identifier (ID) of the VLSM parent of the space where is located the network. 0 indicates that the space where is located the network has no parent space.

vlsm_site_id

The database identifier (ID) of the VLSM child space where the network is duplicated as a VLSM block-type network. 0 indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlsm_site_name

The name of the VLSM child space where the network is duplicated as a VLSM block-type network. 0 indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlmvlan_vlan_id

The VLAN identifier (ID) of the VLAN associated with the network.

vlmvlan_name

The name of the VLAN associated with the network.

vlmdomain_id

The database identifier (ID) of the VLAN domain associated with the network.

vlmdomain_name

The name of the VLAN domain associated with the network.

vlmrang_id

The database identifier (ID) of the VLAN range associated with the network.

vlmrang_name

The name of the VLAN range associated with the network.

trace_creation_date

The creation date of the IPv6 network, in decimal UNIX date format.

trace_last_update_date

The last time the IPv6 network was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv6 network.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv6 network.

trace_creation_origin

The name of the module where the IPv6 network addition originated.

trace_creation_exec_stack

The call stack of the IPv6 network operation details, as follows: <service1>&<service2>&<service3>... .

trace_creation_usr_login

The login of the user who added the IPv6 network.

trace_creation_origin_usr_login

The login of the user who requested the IPv6 network.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 7.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

subnet6_class_parameters

The class parameters applied to the IPv6 network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

subnet6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&... .

parent_subnet6_class_parameters

The class parameters applied to the parent IPv6 network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

parent_subnet6_class_parameters_properties

The inheritance and/or propagation properties of the class parameters returned in the parameter **parent_subnet6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>.f

Example

In the example below, we call the service **ip6_block6_subnet6_list** with Ruby (NET::Http) using the clause *WHERE* to list all the IPv6 terminal networks.

Example 7.1. Calling the service ip6_block6_subnet6_list using Ruby and WHERE

```
require 'uri'
require 'net/http'

url = URI("https://solid.intranet/rest/ip6_block6_subnet6_list?WHERE%2Fis_terminal=1")

http = Net::HTTP.new(url.host, url.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE

request = Net::HTTP::Get.new(url)
request["x-ipm-username"] = 'aXBtYWRtaW4='
request["x-ipm-password"] = 'YWRtaW4='
request["cache-control"] = 'no-cache'

response = http.request(request)
puts response.read_body
```

Name

ip6_block6_subnet6_info — Display the properties of an IPv6 block/subnet-type network

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

subnet6_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

subnet6_id

The database identifier (ID) of the IPv6 network, a unique numeric key value automatically incremented when you add an IPv6 network. Use the ID to specify the IPv6 network of your choice.

Output Parameters

type

The type of the network.

subnet6_id

The database identifier (ID) of the IPv6 network.

start_ip6_addr

The first IP address of the IPv6 network, in hexadecimal format.

start_hostaddr

The human readable version of the parameter `start_ip6_addr`.

end_ip6_addr

The last IP address of the IPv6 network, in hexadecimal format.

end_hostaddr

The human readable version of the parameter **end_ip6_addr**.

subnet6_name

The name of the IPv6 network.

subnet_size

The number of IP addresses the IPv6 network contains.

vlsm_block6_id

The database identifier (ID) of the IPv6 VLSM block-type network duplicated, in a VLSM child space, from the network. *0* indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlmvlan_id

The database identifier (ID) of the VLAN associated with the network.

subnet_level

The level of the network within the space. It returns values between *0* (block-type network) and *n* (subnet-type network). A value higher than *1* indicates a VLSM organization where a block-type network can belong to another subnet-type network.

subnet_path

The path toward the network in the database from the containing block-type network down to the subnet-type network: <block-network-start-IP>#<block-network-ID>#<subnet-network-start-IP>#<subnet-network-ID>. The IP address is returned in hexadecimal format.

- In network-based VLSM organizations, the path includes all the subnet-type networks there are from the containing block-type network down to the subnet-type network specified in *subnet_id*.
- In space-based VLSM organizations, the path includes the block-type network of the top parent space and all the subnet-type networks there are until the network specified in *subnet_id*. Only one block-type network is returned.

subnet6_class_name

The name of the class applied to the IPv6 network, it can be preceded by the class directory.

parent_subnet6_id

The database identifier (ID) of the parent IPv6 network. *0* indicates that the network has no parent network.

vlsm_subnet6_id

The database identifier (ID) of the IPv6 subnet-type network, located in the VLSM parent space, from which the network was duplicated. *0* indicates that the network is not a VLSM block-type network duplicated from a parent space.

row_enabled

The object activation status:

- *0* indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- *1* indicates the object is enabled and managed.
- *2* indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to *1* when an object is created.

subnet6_is_valid

The network validity. A valid network (*1*) has a prefix and last IP address that match.

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

waiting_state

The state of the exchange between SOLIDserver and the RIPE for the assigned network:

Table 7.3. waiting_state possible values

| Status | Description |
|--------------------|---|
| must_send_mail_add | An email must be sent to the RIPE to notify them of a subnet-type network creation. |
| wait_mail_add | A network creation email was sent to the RIPE, no reply has been received yet. |
| must_send_mail_del | An email must be sent to the RIPE to notify them of a subnet-type network deletion. |
| wait_mail_del | A network deletion email was sent to the RIPE, no reply has been received yet. |
| wait_aw_confirm | The number of IP addresses of the assigned network exceeds the Assignment Window declared during your RIPE configuration. |

waiting_status

The status of a RIPE assigned network within SOLIDserver until it is confirmed that you can create or delete it. If set to 1, it is about to be created. If set to 2, it is about to be deleted.

is_terminal

A way to determine if a network can contain other networks. If set to 1, the network is terminal and cannot contain other subnet-type networks. Block-type networks are always set to 0.

subnet6_prefix

The prefix of the IPv6 network.

percent_allocated

The percentage of subnet-type networks the non-terminal network contains.

percent_used

The percentage of terminal networks the non-terminal network contains.

is_in_orphan

A way to determine if the network has a parent (0) or if it belongs to a container *Orphan networks* (1).

site_description

The description of the space the object belongs to.

site_name

The name of the space the object belongs to.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

tree_level

The database level of the space the object belongs to. If you set up a VLSM organization, it returns values between between 0 (the highest level) and *n*.

tree_path

The database path toward the space the object belongs to as follows: <space-name># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#... .

tree_id_path

The database path toward the space the object belongs to as follows: <space-ID>#. If you set up a VLSM organization, the path looks as follows: <highest-level-space-ID>#<child-space-ID>#<child-space-ID>#... .

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

parent_subnet6_name

The name of the parent IPv6 network. # indicates that the network has no parent network.

parent_start_ip6_addr

The first IP address of the parent IPv6 network, in hexadecimal format.

parent_end_ip6_addr

The last IP address of the parent IPv6 network, in hexadecimal format.

parent_subnet_size

The number of IP addresses of the network parent, in hexadecimal format.

parent_subnet_level

The level of the parent network within the space. It returns values between 0 (block-type network) and n (subnet-type network). A value higher than 1 indicates a VLSM organization where a block-type network can belong to another subnet-type network.

parent_subnet_path

The path toward the parent network in the database. # indicates the network has no parent network.

parent_subnet6_class_name

The name of the class applied to the parent IPv6 network, it can be preceded by the class directory.

parent_is_terminal

A way to determine if the parent network is terminal (1) or non-terminal (0).

parent_vlsm_subnet6_id

The database identifier (ID) of the IPv6 subnet-type network, located in the VLSM parent space, from which the parent network was duplicated. 0 indicates that the parent network is not a VLSM block-type network duplicated from a parent space.

parent_subnet6_prefix

The prefix of the parent of the IPv6 network the object belongs to.

parent_percent_allocated

The percentage of subnet-type networks the parent network contains.

parent_percent_used

The percentage of terminal networks the parent network contains.

parent_site_id

The database identifier (ID) of the space where is located the parent network. 0 indicates that the network has no parent network.

parent_site_name

The name of the space where is located the parent network. # indicates that the network has no parent network.

site_parent_site_id

The database identifier (ID) of the VLSM parent of the space where is located the network. 0 indicates that the space where is located the network has no parent space.

vlsm_site_id

The database identifier (ID) of the VLSM child space where the network is duplicated as a VLSM block-type network. 0 indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlsm_site_name

The name of the VLSM child space where the network is duplicated as a VLSM block-type network. 0 indicates that the network is not duplicated as a VLSM block-type network in a child space.

vlmvlan_vlan_id

The VLAN identifier (ID) of the VLAN associated with the network.

vlmvlan_name

The name of the VLAN associated with the network.

vlmdomain_id

The database identifier (ID) of the VLAN domain associated with the network.

vlmdomain_name

The name of the VLAN domain associated with the network.

vlmrange_id

The database identifier (ID) of the VLAN range associated with the network.

vlmrange_name

The name of the VLAN range associated with the network.

trace_creation_date

The creation date of the IPv6 network, in decimal UNIX date format.

trace_last_update_date

The last time the IPv6 network was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv6 network.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv6 network.

trace_creation_origin

The name of the module where the IPv6 network addition originated.

trace_creation_exec_stack

The call stack of the IPv6 network operation details, as follows: <service1>&<service2>&<service3>... .

trace_creation_usr_login

The login of the user who added the IPv6 network.

trace_creation_origin_usr_login

The login of the user who requested the IPv6 network.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 7.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

subnet6_class_parameters

The class parameters applied to the IPv6 network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

subnet6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

subnet6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&... .

parent_subnet6_class_parameters

The class parameters applied to the parent IPv6 network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

parent_subnet6_class_parameters_properties

The inheritance and/or propagation properties of the class parameters returned in the parameter **parent_subnet6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>.f

Name

ip6_block6_subnet6_count — Count the number of IPv6 block/subnet-type networks

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

ip6_find_free_subnet6 — List the 10 first free IPv6 subnet-type networks

Description

This service allows you to list the 10 first free IPv6 subnet-type networks, terminal or non terminal.

You must execute the service using **rpc**.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(prefix)

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

prefix

The prefix of the subnet-type network(s) you are looking for, an integer that defines the number of addresses the network contains.

| Type | IPv6 prefix (integer between 1 and 128) | Maximum length | N/A |
|---------------|---|----------------|-----|
| Default value | N/A | Can be edited | Yes |

max_find

The maximum number of IPv6 networks to be returned by the service. You can use it to return more than 10 results.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

begin_addr

The first IPv6 address of the range of addresses where you are looking for free networks.

| Type | IPv6 address | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

end_addr

The last IPv6 address of the range of addresses where you are looking for free networks.

| Type | IPv6 address | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

| Type | | Maximum length | N/A |
|---------------|-----|----------------|-----|
| Default value | N/A | Can be edited | Yes |

block6_id

The database identifier (ID) of an existing non-terminal IPv6 network. Use the ID to specify the IPv6 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

use_searched_path

A way to filter the search for subnet-type networks based on the specified **block_id**. If set to 0, the service returns free subnet-type networks within the specified block-type network. If set to 1, the service returns free subnet-type networks within the specified block-type network and within all the non-terminal subnet-type networks it might contain.

| Type | Boolean: 0 1 | Maximum length | N/A |
|---------------|-----------------|----------------|-----|
| Default value | 0 | Can be edited | Yes |

Output Parameters

start_ip6_addr

The first IP address of the IPv6 network, in hexadecimal format.

start_hostaddr6

The first IP address of the IPv6 network, in human readable format.

block6_name

The name of the non-terminal IPv6 network the free subnet-type network(s) belongs to.

cost

An integer between 0 and n that evaluates the best range of IP addresses within a block-type network to create a subnet-type network and avoid fragmentation. The lower the cost, the better the position is. The lowest costs are always returned first.

block6_id

The database identifier (ID) of the non-terminal IPv6 network the free subnet-type network(s) belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause **WHERE**.

Name

ip6_block6_subnet6_groupby — Group IPv6 block/subnet-type networks by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. If you specify several parameters they must be separated by a comma as follows: `GROUPBY=<param1>,<param2>,...`. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter `offset` must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter `limit` must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement `SELECT` is returned.

Name

ip6_block6_subnet6_groupby_count—Count the number of IPv6 block/subnet-type networks grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

group_subnet6_add — Add an IPv6 block/subnet-type network to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (subnet6_id || (subnet6_addr && (subnet6_end_addr || subnet6_prefix) && (site_id || site_name || parent_subnet6_id))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet6_id

The database identifier (ID) of the IPv6 network, a unique numeric key value automatically incremented when you add an IPv6 network. Use the ID to specify the IPv6 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|------|-------------|----------------|-----|
|------|-------------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

parent_subnet6_id

The database identifier (ID) of the parent IPv6 network. Use the ID to specify the parent IPv6 network of your choice.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet6_name

The name of the IPv6 network.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

subnet6_addr

The start IP address of the IPv6 network, its first IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet6_end_addr

The end IP address of the IPv6 network, its last IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet6_prefix

The prefix of the IPv6 network, an integer that defines the number of address the network contains.

| | | | |
|----------------------|---|-----------------------|-----|
| Type | IPv6 prefix (integer between 1 and 128) | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_level

The level of the network within the space:

- Set it to *0* for a block-type network.
- Set it to a value between *1* and *n* for a subnet-type network.

If you set a value between *2* and *n*, you are setting a network-based VLSM organization where non terminal subnet-type networks can contain other subnet-type networks.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

relative_position

The position of the network within the hierarchy of networks of a VLSM organization. It calculates between *0* and *n* all the levels of the organization, its behavior depends on the value of the parameter **use_reversed_relative_position**:

- *use_reversed_relative_position=0* where *0* indicates a block-type network at the highest level possible, in a space-based organization, it belongs to the top space. The levels increment from *0* down to *n*, the lowest level you set up, within networks or spaces.

- *use_reversed_relative_position=1* where 1 indicates a network located at the lowest level of the organization, within networks or spaces. The levels increment from 0 up to *n*, the network at the highest level of the organization.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

use_reversed_relative_position

A way to determine if the calculation of the parameter **relative_position** should start from the top (0) or the bottom (1) of the VLSM organization.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

group_subnet6_delete — Remove an IPv6 block/subnet-type network from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (subnet6_id || (subnet6_addr && (subnet6_end_addr || subnet6_prefix) && (site_id || site_name || parent_subnet6_id))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet6_id

The database identifier (ID) of the IPv6 network, a unique numeric key value automatically incremented when you add an IPv6 network. Use the ID to specify the IPv6 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|------|-------------|----------------|-----|
| | | | |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

parent_subnet6_id

The database identifier (ID) of the parent IPv6 network. Use the ID to specify the parent IPv6 network of your choice.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet6_name

The name of the IPv6 network.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

subnet6_addr

The start IP address of the IPv6 network, its first IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet6_end_addr

The end IP address of the IPv6 network, its last IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet6_prefix

The prefix of the IPv6 network, an integer that defines the number of address the network contains.

| | | | |
|----------------------|---|-----------------------|-----|
| Type | IPv6 prefix (integer between 1 and 128) | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_level

The level of the network within the space:

- Set it to *0* for a block-type network.
- Set it to a value between *1* and *n* for a subnet-type network.

If you set a value between *2* and *n*, you are setting a network-based VLSM organization where non terminal subnet-type networks can contain other subnet-type networks.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

relative_position

The position of the network within the hierarchy of networks of a VLSM organization. It calculates between *0* and *n* all the levels of the organization, its behavior depends on the value of the parameter **use_reversed_relative_position**:

- *use_reversed_relative_position=0* where *0* indicates a block-type network at the highest level possible, in a space-based organization, it belongs to the top space. The levels increment from *0* down to *n*, the lowest level you set up, within networks or spaces.

- *use_reversed_relative_position=1* where 1 indicates a network located at the lowest level of the organization, within networks or spaces. The levels increment from 0 up to *n*, the network at the highest level of the organization.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

use_reversed_relative_position

A way to determine if the calculation of the parameter **relative_position** should start from the top (0) or the bottom (1) of the VLSM organization.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

ip6_subnet6_delete — Delete an IPv6 block/subnet-type network

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(subnet6_id || (subnet6_addr && (subnet6_end_addr || subnet6_prefix) && (site_id || site_name || parent_subnet6_id)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet6_id

The database identifier (ID) of the IPv6 network, a unique numeric key value automatically incremented when you add an IPv6 network. Use the ID to specify the IPv6 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

parent_subnet6_id

The database identifier (ID) of the parent IPv6 network. Use the ID to specify the parent IPv6 network of your choice.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet6_name

The name of the IPv6 network.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet6_addr

The start IP address of the IPv6 network, its first IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

subnet6_end_addr

The end IP address of the IPv6 network, its last IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet6_prefix

The prefix of the IPv6 network, an integer that defines the number of address the network contains.

| | | | |
|---------------|---|----------------|-----|
| Type | IPv6 prefix (integer between 1 and 128) | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_level

The level of the network within the space:

- Set it to 0 for a block-type network.
- Set it to a value between 1 and n for a subnet-type network.

If you set a value between 2 and n , you are setting a network-based VLSM organization where non terminal subnet-type networks can contain other subnet-type networks.

| | | | |
|---------------|------------------|----------------|-----|
| Type | Integer ≥ 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

relative_position

The position of the network within the hierarchy of networks of a VLSM organization. It calculates between 0 and n all the levels of the organization, its behavior depends on the value of the parameter **use_reversed_relative_position**:

- *use_reversed_relative_position=0* where 0 indicates a block-type network at the highest level possible, in a space-based organization, it belongs to the top space. The levels increment from 0 down to n , the lowest level you set up, within networks or spaces.
- *use_reversed_relative_position=1* where 1 indicates a network located at the lowest level of the organization, within networks or spaces. The levels increment from 0 up to n , the network at the highest level of the organization.

| | | | |
|---------------|------------------|----------------|-----|
| Type | Integer ≥ 0 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

use_reversed_relative_position

A way to determine if the calculation of the parameter **relative_position** should start from the top (0) or the bottom (1) of the VLSM organization.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 8. IPv4 Pool

Name

ip_pool_add — Add/Edit an IPv4 pool

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (start_addr && (end_addr || pool_size) && (subnet_id || site_id || site_name))
- **Editing:** (pool_id || (start_addr && (end_addr || pool_size) && (subnet_id || site_id || site_name)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet_id

The database identifier (ID) of the IPv4 network, a unique numeric key value automatically incremented when you add an IPv4 network. Use the ID to specify the IPv4 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

pool_id

The database identifier (ID) of the IPv4 pool, a unique numeric key value automatically incremented when you add an IPv4 pool. Use the ID to specify the IPv4 pool of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

pool_name

The name of the IPv4 pool, each IPv4 pool must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

start_addr

The first IP address of the pool.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

The last IP address of the pool.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool_size

The size of the pool, the number of IP addresses it contains.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

pool_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

pool_read_only

The reservation status of the IPv4 pool. If set 1, the IP addresses it contains cannot be assigned.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>.class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... . If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.

- *Warning:* the service execution can continue but an issue might have occurred.
- *Notice:* the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

ip_pool_list — List the IPv4 pools

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

site_name

The name of the space the object belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

site_description

The description of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

tree_path

The database path toward the space the object belongs to as follows: <space-name># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#... .

pool_id

The database identifier (ID) of the IPv4 pool.

pool_name

The name of the IPv4 pool.

pool_class_name

The name of the class applied to the IPv4 pool, it can be preceded by the class directory.

pool_read_only

The reservation status of the IPv4 pool. If set 1, the IP addresses it contains cannot be assigned.

start_ip_addr

The first IP address of the IPv4 pool, in hexadecimal format.

start_hostaddr

The human readable version of the parameter **start_ip_addr**.

end_ip_addr

The last IP address of the IPv4 pool, in hexadecimal format.

end_hostaddr

The human readable version of the parameter **end_ip_addr**.

pool_start_ip_addr

The first IP address of the IPv4 pool, in hexadecimal format.

pool_end_ip_addr

The last IP address of the IPv4 pool, in hexadecimal format.

pool_size

The number of IP addresses the IPv4 pool contains.

parent_subnet_name

The name of the parent IPv4 network:

- # indicates that the network the object belongs to has no parent network.
- *Default* indicates that the network the object belongs to is in an orphan network.

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. It identifies the parent of the IPv4 network the object belongs to. 0 indicates that the network the object belongs to has no parent network.

parent_subnet_size

The number of IP addresses of the parent of the network the object belongs to.

vlsm_subnet_id

The database identifier (ID) of the IPv4 subnet-type network, located in the VLSM parent space, from which the parent of the network the pool belongs to was duplicated. 0 indicates that the parent of the network the pool belongs to is not a VLSM block-type network duplicated from a parent space.

parent_subnet_class_name

The name of the class applied to the parent of the IPv4 network the object belongs to, it can be preceded by the class directory.

subnet_name

The name of the IPv4 network the object belongs to. *Default* indicates that the network the object belongs to is an orphan network.

vlsm_block_id

The database identifier (ID) of the IPv4 VLSM block-type network duplicated, in a VLSM child space, from the network the pool belongs to. 0 indicates that the parent of the network the pool belongs to is not duplicated as a VLSM block-type network in a child space.

subnet_id

The database identifier (ID) of the IPv4 network the object belongs to.

subnet_start_ip_addr

The first IP address of the IPv4 network the object belongs to.

subnet_end_ip_addr

The last IP address of the IPv4 network the object belongs to.

subnet_size

The number of IP addresses the network the object belongs to contains.

subnet_class_name

The name of the class applied to the IPv4 network the object belongs to, it can be preceded by the class directory.

trace_creation_date

The creation date of the IPv4 pool, in decimal UNIX date format.

trace_last_update_date

The last time the IPv4 pool was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv4 pool.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv4 pool.

trace_creation_origin

The name of the module where the IPv4 pool addition originated.

trace_creation_exec_stack

The call stack of the IPv4 pool operation details, as follows: <service1>&<service2>&<service3>....

trace_creation_usr_login

The login of the user who added the IPv4 pool.

trace_creation_origin_usr_login

The login of the user who requested the IPv4 pool.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 8.1. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

pool_class_parameters

The class parameters applied to the IPv4 pool and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

pool_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **pool_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

pool_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
site_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&... .

subnet_class_parameters

The class parameters applied to the IPv4 network the object belongs to and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
subnet_class_parameters: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&... .

Name

ip_pool_info — Display the properties of an IPv4 pool

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

pool_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

pool_id

The database identifier (ID) of the IPv4 pool, a unique numeric key value automatically incremented when you add an IPv4 pool. Use the ID to specify the IPv4 pool of your choice.

Output Parameters

site_name

The name of the space the object belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

site_description

The description of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

tree_path

The database path toward the space the object belongs to as follows: <space-name># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#....

pool_id

The database identifier (ID) of the IPv4 pool.

pool_name

The name of the IPv4 pool.

pool_class_name

The name of the class applied to the IPv4 pool, it can be preceded by the class directory.

pool_read_only

The reservation status of the IPv4 pool. If set 1, the IP addresses it contains cannot be assigned.

start_ip_addr

The first IP address of the IPv4 pool, in hexadecimal format.

start_hostaddr

The human readable version of the parameter **start_ip_addr**.

end_ip_addr

The last IP address of the IPv4 pool, in hexadecimal format.

end_hostaddr

The human readable version of the parameter **end_ip_addr**.

pool_start_ip_addr

The first IP address of the IPv4 pool, in hexadecimal format.

pool_end_ip_addr

The last IP address of the IPv4 pool, in hexadecimal format.

pool_size

The number of IP addresses the IPv4 pool contains.

parent_subnet_name

The name of the parent IPv4 network:

- # indicates that the network the object belongs to has no parent network.
- *Default* indicates that the network the object belongs to is in an orphan network.

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. It identifies the parent of the IPv4 network the object belongs to. 0 indicates that the network the object belongs to has no parent network.

parent_subnet_size

The number of IP addresses of the parent of the network the object belongs to.

vlsm_subnet_id

The database identifier (ID) of the IPv4 subnet-type network, located in the VLSM parent space, from which the parent of the network the pool belongs to was duplicated. 0 indicates that the parent of the network the pool belongs to is not a VLSM block-type network duplicated from a parent space.

parent_subnet_class_name

The name of the class applied to the parent of the IPv4 network the object belongs to, it can be preceded by the class directory.

subnet_name

The name of the IPv4 network the object belongs to. *Default* indicates that the network the object belongs to is an orphan network.

vlsm_block_id

The database identifier (ID) of the IPv4 VLSM block-type network duplicated, in a VLSM child space, from the network the pool belongs to. *0* indicates that the parent of the network the pool belongs to is not duplicated as a VLSM block-type network in a child space.

subnet_id

The database identifier (ID) of the IPv4 network the object belongs to.

subnet_start_ip_addr

The first IP address of the IPv4 network the object belongs to.

subnet_end_ip_addr

The last IP address of the IPv4 network the object belongs to.

subnet_size

The number of IP addresses the network the object belongs to contains.

subnet_class_name

The name of the class applied to the IPv4 network the object belongs to, it can be preceded by the class directory.

trace_creation_date

The creation date of the IPv4 pool, in decimal UNIX date format.

trace_last_update_date

The last time the IPv4 pool was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv4 pool.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv4 pool.

trace_creation_origin

The name of the module where the IPv4 pool addition originated.

trace_creation_exec_stack

The call stack of the IPv4 pool operation details, as follows: <service1>&<service2>&<service3>... .

trace_creation_usr_login

The login of the user who added the IPv4 pool.

trace_creation_origin_usr_login

The login of the user who requested the IPv4 pool.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 8.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

pool_class_parameters

The class parameters applied to the IPv4 pool and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

pool_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **pool_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

pool_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&.... .

subnet_class_parameters

The class parameters applied to the IPv4 network the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet_class_parameters**: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&.... .

Name

ip_pool_count — Count the number of IPv4 pools

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

group_pool_add — Add an IPv4 pool to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (pool_id || (start_addr && (end_addr || pool_size) && (subnet_id || site_id || site_name))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet_id

The database identifier (ID) of the IPv4 network, a unique numeric key value automatically incremented when you add an IPv4 network. Use the ID to specify the IPv4 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|------|-------------|----------------|-----|
|------|-------------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

pool_id

The database identifier (ID) of the IPv4 pool, a unique numeric key value automatically incremented when you add an IPv4 pool. Use the ID to specify the IPv4 pool of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

start_addr

The first IP address of the pool.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

The last IP address of the pool.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool_size

The size of the pool, the number of IP addresses it contains.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

group_pool_delete — Remove an IPv4 pool from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (pool_id || (start_addr && (end_addr || pool_size) && (subnet_id || site_id || site_name))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

subnet_id

The database identifier (ID) of the IPv4 network, a unique numeric key value automatically incremented when you add an IPv4 network. Use the ID to specify the IPv4 network of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

pool_id

The database identifier (ID) of the IPv4 pool, a unique numeric key value automatically incremented when you add an IPv4 pool. Use the ID to specify the IPv4 pool of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

start_addr

The first IP address of the pool.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

The last IP address of the pool.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool_size

The size of the pool, the number of IP addresses it contains.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

ip_pool_delete — Delete an IPv4 pool

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(pool_id || (start_addr && (end_addr || pool_size) && (subnet_id || site_id || site_name)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

subnet_id

The database identifier (ID) of the IPv4 network, a unique numeric key value automatically incremented when you add an IPv4 network. Use the ID to specify the IPv4 network of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool_id

The database identifier (ID) of the IPv4 pool, a unique numeric key value automatically incremented when you add an IPv4 pool. Use the ID to specify the IPv4 pool of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

start_addr

The first IP address of the pool.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

The last IP address of the pool.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool_size

The size of the pool, the number of IP addresses it contains.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 9. IPv6 Pool

Name

ip6_pool6_add — Add/Edit an IPv6 pool

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (start_addr && end_addr && (subnet6_id || site_id || site_name))
- **Editing:** (pool6_id || (start_addr && end_addr && (subnet6_id || site_id || site_name)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet6_id

The database identifier (ID) of the IPv6 network, a unique numeric key value automatically incremented when you add an IPv6 network. Use the ID to specify the IPv6 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

pool6_id

The database identifier (ID) of the IPv6 pool, a unique numeric key value automatically incremented when you add an IPv6 pool. Use the ID to specify the IPv6 pool of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

pool6_name

The name of the IPv6 pool, each IPv6 pool must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

start_addr

The first IP address of the pool.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

The last IP address of the pool.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool6_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

pool6_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

pool6_read_only

The reservation status of the IPv6 pool. If set 1, the IP addresses it contains cannot be assigned.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool6_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>_class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

ip6_pool6_list — List the IPv6 pools

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

site_name

The name of the space the object belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

site_description

The description of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

tree_path

The database path toward the space the object belongs to as follows: <space-name># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#... .

pool6_id

The database identifier (ID) of the IPv6 pool.

pool6_name

The name of the IPv6 pool.

pool6_class_name

The name of the class applied to the IPv6 pool, it can be preceded by the class directory.

pool6_read_only

The reservation status of the IPv6 pool. If set 1, the IP addresses it contains cannot be assigned.

start_ip6_addr

The first IP address of the IPv6 pool, in hexadecimal format.

start_hostaddr

The human readable version of the parameter **start_ip6_addr**.

end_ip6_addr

The last IP address of the IPv6 pool, in hexadecimal format.

end_hostaddr

The human readable version of the parameter **end_ip6_addr**.

pool6_start_ip6_addr

The first IP address of the IPv6 pool, in hexadecimal format.

pool6_end_ip6_addr

The last IP address of the IPv6 pool, in hexadecimal format.

pool6_size

The number of IP addresses the IPv6 pool contains.

parent_subnet6_name

The name of the parent IPv6 network. # indicates that the network the object belongs to has no parent network.

parent_subnet6_id

The database identifier (ID) of the parent IPv6 network. It identifies the parent of the IPv6 network the object belongs to. 0 indicates that the network the object belongs to has no parent network.

vlsm_subnet6_id

The database identifier (ID) of the IPv6 subnet-type network, located in the VLSM parent space, from which the parent of the network the pool belongs to was duplicated. 0 indicates that the parent of the network the pool belongs to is not a VLSM block-type network duplicated from a parent space.

parent_subnet6_prefix

The prefix of the parent of the IPv6 network the object belongs to.

parent_subnet6_class_name

The name of the class applied to the parent of the IPv6 network the object belongs to, it can be preceded by the class directory.

subnet6_name

The name of the IPv6 network the object belongs to.

vlsm_block6_id

The database identifier (ID) of the IPv6 VLSM block-type network duplicated, in a VLSM child space, from the network the pool belongs to. 0 indicates that the parent of the network the pool belongs to is not duplicated as a VLSM block-type network in a child space.

subnet6_id

The database identifier (ID) of the IPv6 network the object belongs to.

subnet6_start_ip6_addr

The first IP address of the IPv6 network the object belongs to.

subnet6_end_ip6_addr

The last IP address of the IPv6 network the object belongs to.

subnet6_class_name

The name of the class applied to the IPv6 network the object belongs to, it can be preceded by the class directory.

subnet6_prefix

The prefix of the IPv6 network the object belongs to.

trace_creation_date

The creation date of the IPv6 pool, in decimal UNIX date format.

trace_last_update_date

The last time the IPv6 pool was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv6 pool.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv6 pool.

trace_creation_origin

The name of the module where the IPv6 pool addition originated.

trace_creation_exec_stack

The call stack of the IPv6 pool operation details, as follows: <service1>&<service2>&<service3>... .

trace_creation_usr_login

The login of the user who added the IPv6 pool.

trace_creation_origin_usr_login

The login of the user who requested the IPv6 pool.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 9.1. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

pool6_class_parameters

The class parameters applied to the IPv6 pool and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

pool6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **pool6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

pool6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&... .

subnet6_class_parameters

The class parameters applied to the IPv6 network the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

Name

ip6_pool6_info — Display the properties of an IPv6 pool

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

pool6_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

pool6_id

The database identifier (ID) of the IPv6 pool, a unique numeric key value automatically incremented when you add an IPv6 pool. Use the ID to specify the IPv6 pool of your choice.

Output Parameters

site_name

The name of the space the object belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

site_description

The description of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

tree_path

The database path toward the space the object belongs to as follows: <space-name># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#....

pool6_id

The database identifier (ID) of the IPv6 pool.

pool6_name

The name of the IPv6 pool.

pool6_class_name

The name of the class applied to the IPv6 pool, it can be preceded by the class directory.

pool6_read_only

The reservation status of the IPv6 pool. If set 1, the IP addresses it contains cannot be assigned.

start_ip6_addr

The first IP address of the IPv6 pool, in hexadecimal format.

start_hostaddr

The human readable version of the parameter **start_ip6_addr**.

end_ip6_addr

The last IP address of the IPv6 pool, in hexadecimal format.

end_hostaddr

The human readable version of the parameter **end_ip6_addr**.

pool6_start_ip6_addr

The first IP address of the IPv6 pool, in hexadecimal format.

pool6_end_ip6_addr

The last IP address of the IPv6 pool, in hexadecimal format.

pool6_size

The number of IP addresses the IPv6 pool contains.

parent_subnet6_name

The name of the parent IPv6 network. # indicates that the network the object belongs to has no parent network.

parent_subnet6_id

The database identifier (ID) of the parent IPv6 network. It identifies the parent of the IPv6 network the object belongs to. 0 indicates that the network the object belongs to has no parent network.

vlsm_subnet6_id

The database identifier (ID) of the IPv6 subnet-type network, located in the VLSM parent space, from which the parent of the network the pool belongs to was duplicated. 0 indicates that the parent of the network the pool belongs to is not a VLSM block-type network duplicated from a parent space.

parent_subnet6_prefix

The prefix of the parent of the IPv6 network the object belongs to.

parent_subnet6_class_name

The name of the class applied to the parent of the IPv6 network the object belongs to, it can be preceded by the class directory.

subnet6_name

The name of the IPv6 network the object belongs to.

vlsm_block6_id

The database identifier (ID) of the IPv6 VLSM block-type network duplicated, in a VLSM child space, from the network the pool belongs to. 0 indicates that the parent of the network the pool belongs to is not duplicated as a VLSM block-type network in a child space.

subnet6_id

The database identifier (ID) of the IPv6 network the object belongs to.

subnet6_start_ip6_addr

The first IP address of the IPv6 network the object belongs to.

subnet6_end_ip6_addr

The last IP address of the IPv6 network the object belongs to.

subnet6_class_name

The name of the class applied to the IPv6 network the object belongs to, it can be preceded by the class directory.

subnet6_prefix

The prefix of the IPv6 network the object belongs to.

trace_creation_date

The creation date of the IPv6 pool, in decimal UNIX date format.

trace_last_update_date

The last time the IPv6 pool was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv6 pool.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv6 pool.

trace_creation_origin

The name of the module where the IPv6 pool addition originated.

trace_creation_exec_stack

The call stack of the IPv6 pool operation details, as follows: <service1>&<service2>&<service3>... .

trace_creation_usr_login

The login of the user who added the IPv6 pool.

trace_creation_origin_usr_login

The login of the user who requested the IPv6 pool.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 9.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

pool6_class_parameters

The class parameters applied to the IPv6 pool and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

pool6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **pool6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

pool6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&... .

subnet6_class_parameters

The class parameters applied to the IPv6 network the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

subnet6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

Name

ip6_pool6_count — Count the number of IPv6 pools

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

group_pool6_add — Add an IPv6 pool to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (pool6_id || (start_addr && end_addr && (subnet6_id || site_id || site_name))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet6_id

The database identifier (ID) of the IPv6 network, a unique numeric key value automatically incremented when you add an IPv6 network. Use the ID to specify the IPv6 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|------|-------------|----------------|-----|
| | | | |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

pool6_id

The database identifier (ID) of the IPv6 pool, a unique numeric key value automatically incremented when you add an IPv6 pool. Use the ID to specify the IPv6 pool of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

start_addr

The first IP address of the pool.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

The last IP address of the pool.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

group_pool6_delete — Remove an IPv6 pool from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (pool6_id || (start_addr && end_addr && (subnet6_id || site_id || site_name))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet6_id

The database identifier (ID) of the IPv6 network, a unique numeric key value automatically incremented when you add an IPv6 network. Use the ID to specify the IPv6 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|------|-------------|----------------|-----|
|------|-------------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

pool6_id

The database identifier (ID) of the IPv6 pool, a unique numeric key value automatically incremented when you add an IPv6 pool. Use the ID to specify the IPv6 pool of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

start_addr

The first IP address of the pool.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

The last IP address of the pool.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

ip6_pool6_delete — Delete an IPv6 pool

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(pool6_id || (start_addr && end_addr && (subnet6_id || site_id || site_name)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

subnet6_id

The database identifier (ID) of the IPv6 network, a unique numeric key value automatically incremented when you add an IPv6 network. Use the ID to specify the IPv6 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

pool6_id

The database identifier (ID) of the IPv6 pool, a unique numeric key value automatically incremented when you add an IPv6 pool. Use the ID to specify the IPv6 pool of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

start_addr

The first IP address of the pool.

| Type | IPv6 address | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

end_addr

The last IP address of the pool.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 10. IPv4 Address

Name

ip_add — Add/Edit an IPv4 address

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (hostaddr && (site_id || site_name))
- **Editing:** (ip_id || (hostaddr && (site_id || site_name)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip_id

The database identifier (ID) of the IPv4 address, a unique numeric key value automatically incremented when you add an IPv4 address. Use the ID to specify the IPv4 address of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

name

The name of the IPv4 address, each IPv4 address must have a unique name.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | | Can be edited | Yes |

ip_name

Deprecated, replaced by **name**.

mac_addr

The MAC address you want to associate with the IPv4 address.

| | | | |
|---------------|-------------|----------------|-----|
| Type | MAC address | Maximum length | 64 |
| Default value | | Can be edited | Yes |

ip_addr

Deprecated, replaced by **hostaddr**.

hostaddr

The IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

ip_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

ip_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

hostdev_id

The database identifier (ID) of the Device Manager device you want to associate with the IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

hostiface_id

The database identifier (ID) of the Device Manager interface you want to associate with the IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

iplport_id

The database identifier (ID) of the NetChange port you want to associate with the IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dhcphost_id

The database identifier (ID) of the DHCP static you want to associate with the IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | | Can be edited | Yes |

| | | | |
|----------------------|---|----------------------|-----|
| Default value | 0 | Can be edited | Yes |
|----------------------|---|----------------------|-----|

dhcplease_id

The database identifier (ID) of the DHCP lease you want to associate with the IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

check_is_dhcp_ip

A way to force a validity check, if you configured the IPAM to DHCP replication. If the check is enabled (1), the configuration of the IP address you are adding must be valid as well for the DHCP.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ip_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>.class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Example

In the example below, we call the service **ip_add** with Ruby (NET::Http) to an IPv4 address named *ip.domain.corp* in one of our spaces.

Example 10.1. Calling the service ip_add using Ruby

```
require 'uri'
require 'net/http'

url =
URI("https://solid.intranet/rest/ip_add?hostaddr=15.0.0.10&name=ip.domain.corp&site_id=44")

http = Net::HTTP.new(url.host, url.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE

request = Net::HTTP::Post.new(url)
request["x-ipm-username"] = 'aXBtYWRtaW4='
```

```
request[ "x-ipm-password" ] = 'YWRtaW4='  
request[ "cache-control" ] = 'no-cache'  
  
response = http.request(request)  
puts response.read_body
```

Name

ip_address_list — List the IPv4 addresses

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

type

A way to determine if you can assign the IP address (*free*) or if it is In use (*ip*).

free_start_ip_addr

An IP address in hexadecimal format. For addresses *In use* (**type ip**), it returns the IP address itself.

For free addresses (**type free**), it returns the first IP address of a range of IPv4 addresses that are not assigned yet. The last address in that range is returned in *free_end_ip_addr*.

free_end_ip_addr

An IP address in hexadecimal format. For addresses *In use* (**type ip**), it returns the IP address itself.

For free addresses (**type free**), it returns the last IP address of a range of IPv4 addresses that are not assigned yet. The first address in that range is returned in *free_start_ip_addr*.

free_scope_size

The number of IP addresses that are not assigned yet (**type free**) between *free_start_ip_addr* and *free_end_ip_addr*.

ip_id

The database identifier (ID) of the IPv4 address.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

site_name

The name of the space the object belongs to.

tree_level

The database level of the space the object belongs to. If you set up a VLSM organization, it returns values between between 0 (the highest level) and *n*.

tree_path

The database path toward the space the object belongs to as follows: <*space-name*># . If you set up a VLSM organization, the path looks as follows: <*highest-level-space-name*>##<*child-space-name*>#<*child-space-name*>#... .

tree_id_path

The database path toward the space the object belongs to as follows: <*space-ID*># . If you set up a VLSM organization, the path looks as follows: <*highest-level-space-ID*>#<*child-space-ID*>#<*child-space-ID*>#... .

ip_addr

The IPv4 address itself, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr**.

name

The name of the IPv4 address.

mac_addr

The MAC address associated with the IPv4 address.

ip_class_name

The name of the class applied to the IPv4 address, it can be preceded by the class directory.

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. It identifies the parent of the IPv4 network the object belongs to. 0 indicates that the network the object belongs to has no parent network.

parent_subnet_name

The name of the parent IPv4 network:

- # indicates that the network the object belongs to has no parent network.
- *Default* indicates that the network the object belongs to is in an orphan network.

parent_subnet_size

The number of IP addresses of the parent of the network the object belongs to.

parent_vlsm_subnet_id

The database identifier (ID) of the IPv4 subnet-type network, located in the VLSM parent space, from which the parent of the network the IP address belongs to was duplicated. 0 indicates that the parent of the network the IP address belongs to is not a VLSM block-type network duplicated from a parent space.

parent_subnet_class_name

The name of the class applied to the parent of the IPv4 network the object belongs to, it can be preceded by the class directory.

parent_subnet_start_ip_addr

The first IP address of the parent of the IPv4 network the IP address belongs to.

parent_subnet_start_hostaddr

The human readable version of the parameter **parent_subnet_start_ip_addr**.

parent_subnet_end_ip_addr

The last IP address of the parent of the IPv4 network the IP address belongs to.

parent_subnet_end_hostaddr

The human readable version of the parameter **parent_subnet_end_ip_addr**.

subnet_name

The name of the IPv4 network the object belongs to. *Default* indicates that the network the object belongs to is an orphan network.

pool_name

The name of the IPv4 pool the object belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

subnet_id

The database identifier (ID) of the IPv4 network the object belongs to.

subnet_start_ip_addr

The first IP address of the IPv4 network the object belongs to.

subnet_start_hostaddr

The human readable version of the parameter **subnet_start_ip_addr**.

subnet_end_ip_addr

The last IP address of the IPv4 network the object belongs to.

subnet_end_hostaddr

The human readable version of the parameter **subnet_end_ip_addr**.

subnet_size

The number of IP addresses the network the object belongs to contains.

subnet_is_terminal

A way to determine if the network the IP address belongs to is terminal (1) or non-terminal (0).

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network the IP address belongs to.

pool_class_name

The name of the class applied to the IPv4 pool the object belongs to, it can be preceded by the class directory.

pool_id

The database identifier (ID) of the IPv4 pool the object belongs to.

pool_read_only

The reservation status of the pool the IPv4 address belongs to. If set 1, the pool is reserved and you cannot assign the IP address.

pool_row_enabled

Internal use. Not documented.

iplnetdev_name

The name of the NetChange network device associated with the IP address.

iplnetdev_id

The database identifier (ID) of the NetChange network device associated with the IP address.

iplport_name

The name of the NetChange port associated with the IP address.

iplport_slotnumber

The slot number of the port, for IP addresses which MAC addresses is imported from NetChange.

iplport_portnumber

The number of the port, for IP addresses which MAC addresses is imported from NetChange.

iplport_ifvlan

The VLAN identifier (ID) of the NetChange port, for IP addresses which MAC addresses is imported from NetChange.

hostiface_name

The name of the Device Manager interface associated with the IP address.

hostiface_id

The database identifier (ID) of the Device Manager interface associated with the IP address.

hostdev_name

The name of the Device Manager device associated with the IP address.

hostdev_id

The database identifier (ID) of the Device Manager device associated with the IP address.

dhcpghost_id

The database identifier (ID) of the DHCP static associated with the IP address.

dhcplease_id

The database identifier (ID) of the DHCP lease associated with the IP address.

last_seen

The last time the MAC address associated with the IP address was seen on the network, in decimal UNIX date format.

dhcplease_end_time

The expiration time of the lease, if the IP address was imported from the DHCP, in decimal UNIX date format.

site_description

The description of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

subnet_class_name

The name of the class applied to the IPv4 network the object belongs to, it can be preceded by the class directory.

pool_size

The number of IP addresses that contains the pool the IPv4 address belongs to.

pool_start_ip_addr

The first IP address of the IPv4 pool the IP address belongs to.

pool_end_ip_addr

The last IP address of the IPv4 pool the IP address belongs to.

ip_alias

The name of the IPv4 alias(es) associated with the IPv4 address.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 10.1. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|---|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

tag_pool_dhcprange

A way to determine if the pool the IP address belongs to is configured to Create DHCP range (1) or not (0).

tag_container_dhcpstatic

A way to determine if the terminal network or pool the IP address belongs to is configured to Add a DHCP static (1) or not (0).

trace_creation_date

The creation date of the IPv4 address, in decimal UNIX date format.

trace_last_update_date

The last time the IPv4 address was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv4 address.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv4 address.

trace_creation_origin

The name of the module where the IPv4 address addition originated.

trace_creation_exec_stack

The call stack of the IPv4 address operation details, as follows: <service1>&<service2>&<service3>.... .

trace_creation_usr_login

The login of the user who added the IPv4 address.

trace_creation_origin_usr_login

The login of the user who requested the IPv4 address.

ip_class_parameters

The class parameters applied to the IPv4 address and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

ip_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **ip_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

ip_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

pool_class_parameters

The class parameters applied to the IPv4 pool the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

pool_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **pool_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&.... .

subnet_class_parameters

The class parameters applied to the IPv4 network the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet_class_parameters**: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&.... .

Example

In the example below, we call the service **ip_address_list** with PHP (cURL) and the clause *WHERE* to list the only used IP addresses that are not called *Gateway* and are configured with a class called **staff**.

Example 10.2. Calling the service ip_address_list using PHP

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/ip_address_list?WHERE".
    "=type%3D%27ip%27%20and%20name%21%3D%27Gateway%27%20and%20ip_class_name%20like%20%27%25staff%25%27",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Name

ip_address_info — Display the properties of an IPv4 address

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

ip_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

ip_id

The database identifier (ID) of the IPv4 address, a unique numeric key value automatically incremented when you add an IPv4 address. Use the ID to specify the IPv4 address of your choice.

Output Parameters

type

A way to determine if you can assign the IP address (`free`) or if it is In use (`ip`).

free_start_ip_addr

An IP address in hexadecimal format. For addresses `In use (type ip)`, it returns the IP address itself.

For free addresses (`type free`), it returns the first IP address of a range of IPv4 addresses that are not assigned yet. The last address in that range is returned in `free_end_ip_addr`.

free_end_ip_addr

An IP address in hexadecimal format. For addresses `In use (type ip)`, it returns the IP address itself.

For free addresses (**type free**), it returns the last IP address of a range of IPv4 addresses that are not assigned yet. The first address in that range is returned in *free_start_ip_addr*.

free_scope_size

The number of IP addresses that are not assigned yet (**type free**) between *free_start_ip_addr* and *free_end_ip_addr*.

ip_id

The database identifier (ID) of the IPv4 address.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

site_name

The name of the space the object belongs to.

tree_level

The database level of the space the object belongs to. If you set up a VLSM organization, it returns values between between 0 (the highest level) and *n*.

tree_path

The database path toward the space the object belongs to as follows: <space-name># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#... .

tree_id_path

The database path toward the space the object belongs to as follows: <space-ID># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-ID>#<child-space-ID>#<child-space-ID>#... .

ip_addr

The IPv4 address itself, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr**.

name

The name of the IPv4 address.

mac_addr

The MAC address associated with the IPv4 address.

ip_class_name

The name of the class applied to the IPv4 address, it can be preceded by the class directory.

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. It identifies the parent of the IPv4 network the object belongs to. 0 indicates that the network the object belongs to has no parent network.

parent_subnet_name

The name of the parent IPv4 network:

- # indicates that the network the object belongs to has no parent network.
- Default indicates that the network the object belongs to is in an orphan network.

parent_subnet_size

The number of IP addresses of the parent of the network the object belongs to.

parent_vlsm_subnet_id

The database identifier (ID) of the IPv4 subnet-type network, located in the VLSM parent space, from which the parent of the network the IP address belongs to was duplicated. 0 in-

dicates that the parent of the network the IP address belongs to is not a VLSM block-type network duplicated from a parent space.

parent_subnet_class_name

The name of the class applied to the parent of the IPv4 network the object belongs to, it can be preceded by the class directory.

parent_subnet_start_ip_addr

The first IP address of the parent of the IPv4 network the IP address belongs to.

parent_subnet_start_hostaddr

The human readable version of the parameter **parent_subnet_start_ip_addr**.

parent_subnet_end_ip_addr

The last IP address of the parent of the IPv4 network the IP address belongs to.

parent_subnet_end_hostaddr

The human readable version of the parameter **parent_subnet_end_ip_addr**.

subnet_name

The name of the IPv4 network the object belongs to. *Default* indicates that the network the object belongs to is an orphan network.

pool_name

The name of the IPv4 pool the object belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

subnet_id

The database identifier (ID) of the IPv4 network the object belongs to.

subnet_start_ip_addr

The first IP address of the IPv4 network the object belongs to.

subnet_start_hostaddr

The human readable version of the parameter **subnet_start_ip_addr**.

subnet_end_ip_addr

The last IP address of the IPv4 network the object belongs to.

subnet_end_hostaddr

The human readable version of the parameter **subnet_end_ip_addr**.

subnet_size

The number of IP addresses the network the object belongs to contains.

subnet_is_terminal

A way to determine if the network the IP address belongs to is terminal (1) or non-terminal (0).

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network the IP address belongs to.

pool_class_name

The name of the class applied to the IPv4 pool the object belongs to, it can be preceded by the class directory.

pool_id

The database identifier (ID) of the IPv4 pool the object belongs to.

pool_read_only

The reservation status of the pool the IPv4 address belongs to. If set 1, the pool is reserved and you cannot assign the IP address.

pool_row_enabled

Internal use. Not documented.

iplnetdev_name

The name of the NetChange network device associated with the IP address.

iplnetdev_id

The database identifier (ID) of the NetChange network device associated with the IP address.

iplport_name

The name of the NetChange port associated with the IP address.

iplport_slotnumber

The slot number of the port, for IP addresses which MAC addresses is imported from NetChange.

iplport_portnumber

The number of the port, for IP addresses which MAC addresses is imported from NetChange.

iplport_ifvlan

The VLAN identifier (ID) of the NetChange port, for IP addresses which MAC addresses is imported from NetChange.

hostiface_name

The name of the Device Manager interface associated with the IP address.

hostiface_id

The database identifier (ID) of the Device Manager interface associated with the IP address.

hostdev_name

The name of the Device Manager device associated with the IP address.

hostdev_id

The database identifier (ID) of the Device Manager device associated with the IP address.

dhcpghost_id

The database identifier (ID) of the DHCP static associated with the IP address.

dhcplease_id

The database identifier (ID) of the DHCP lease associated with the IP address.

last_seen

The last time the MAC address associated with the IP address was seen on the network, in decimal UNIX date format.

dhcplease_end_time

The expiration time of the lease, if the IP address was imported from the DHCP, in decimal UNIX date format.

site_description

The description of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

subnet_class_name

The name of the class applied to the IPv4 network the object belongs to, it can be preceded by the class directory.

pool_size

The number of IP addresses that contains the pool the IPv4 address belongs to.

pool_start_ip_addr

The first IP address of the IPv4 pool the IP address belongs to.

pool_end_ip_addr

The last IP address of the IPv4 pool the IP address belongs to.

ip_alias

The name of the IPv4 alias(es) associated with the IPv4 address.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 10.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

tag_pool_dhcprange

A way to determine if the pool the IP address belongs to is configured to Create DHCP range (1) or not (0).

tag_container_dhcpstatic

A way to determine if the terminal network or pool the IP address belongs to is configured to Add a DHCP static (1) or not (0).

trace_creation_date

The creation date of the IPv4 address, in decimal UNIX date format.

trace_last_update_date

The last time the IPv4 address was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv4 address.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv4 address.

trace_creation_origin

The name of the module where the IPv4 address addition originated.

trace_creation_exec_stack

The call stack of the IPv4 address operation details, as follows: <service1>&<service2>&<service3>... .

trace_creation_usr_login

The login of the user who added the IPv4 address.

trace_creation_origin_usr_login

The login of the user who requested the IPv4 address.

ip_class_parameters

The class parameters applied to the IPv4 address and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

ip_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **ip_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....

ip_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&....

pool_class_parameters

The class parameters applied to the IPv4 pool the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

pool_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **pool_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&....

subnet_class_parameters

The class parameters applied to the IPv4 network the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

subnet_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet_class_parameters**: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&....

Example

In the example below, we call the service **ip_address_info** with PHP (cURL) to retrieve the properties of an IPv4 address.

Example 10.3. Calling the service ip_address_info using PHP

```
<?php  
  
$curl = curl_init();  
  
curl_setopt_array($curl, array(  
    CURLOPT_URL => "https://solid.intranet/rest/ip_address_info?ip_id=241",  
    CURLOPT_RETURNTRANSFER => true,  
    CURLOPT_ENCODING => "",  
    CURLOPT_MAXREDIRS => 10,
```

```
CURLOPT_TIMEOUT => 30,
CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
CURLOPT_CUSTOMREQUEST => "GET",
CURLOPT_HTTPHEADER => array(
    "cache-control: no-cache",
    "x-ipm-password: YWRtaW4=",
    "x-ipm-username: aXBtYWRtaW4="
),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Name

ip_address_count — Count the number of IPv4 addresses

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

ip_find_free_address — List the free IPv4 addresses

Description

This service allows you to list the 10 first free IPv4 addresses.

You must execute the service using **rpc**.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(subnet_id || pool_id || parent_subnet_id)

Input Parameters

subnet_id

The database identifier (ID) of the IPv4 network, a unique numeric key value automatically incremented when you add an IPv4 network. Use the ID to specify the IPv4 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. Use the ID to specify the parent IPv4 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

pool_id

The database identifier (ID) of the IPv4 pool, a unique numeric key value automatically incremented when you add an IPv4 pool. Use the ID to specify the IPv4 pool of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

max_find

The maximum number of IPv4 addresses to be returned by the service. You can use it to return more than 10 results.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | 10 | Can be edited | Yes |

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

| | | | |
|---------------|-----|----------------|-----|
| Type | | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

begin_addr

The first IPv4 address of the range of addresses where you are looking for free IP addresses.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

The last IPv4 address of the range of addresses where you are looking for free IP addresses.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool_class_name

The name of the class applied to the IPv4 pool the IP addresses you are looking for belong to. You must specify the class file directory, e.g. *my_directory/my_class.class* . You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet_class_name

The name of the class applied to the IPv4 network the IP addresses you are looking for belong to. You must specify the class file directory, e.g. *my_directory/my_class.class* . You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

ip_addr

The IPv4 address itself, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr**.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

site_name

The name of the space the object belongs to.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

subnet_id

The database identifier (ID) of the IPv4 network the object belongs to.

subnet_name

The name of the IPv4 network the object belongs to. *Default* indicates that the network the object belongs to is an orphan network.

pool_id

The database identifier (ID) of the IPv4 pool the object belongs to.

pool_name

The name of the IPv4 pool the object belongs to.

Example

In the example below, we call the service **ip_find_free_address** with Python (Requests) to list the free IPv4 addresses of a specific network. Unlike the other services, it must be called using the method **/rpc/**.

Example 10.4. Calling the service ip_find_free_address using Python

```
import requests

url = "https://solid.intranet/rpc/ip_find_free_address"

querystring = {"subnet_id": "238"}

headers = {
    'x-ipm-username': "aXBtYWRtaW4=",
    'x-ipm-password': "YWRtaW4=",
    'cache-control': "no-cache"
}

response = requests.request("OPTIONS", url, headers=headers, params=querystring)

print(response.text)
```

Name

ip_free_address_list — List the free IPv4 addresses

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

type

A way to determine if you can assign the IP address (*free*) or if it is In use (*ip*).

free_start_ip_addr

An IP address in hexadecimal format. For addresses *In use* (**type ip**), it returns the IP address itself.

For free addresses (**type free**), it returns the first IP address of a range of IPv4 addresses that are not assigned yet. The last address in that range is returned in *free_end_ip_addr*.

free_end_ip_addr

An IP address in hexadecimal format. For addresses *In use* (**type ip**), it returns the IP address itself.

For free addresses (**type free**), it returns the last IP address of a range of IPv4 addresses that are not assigned yet. The first address in that range is returned in *free_start_ip_addr*.

free_scope_size

The number of IP addresses that are not assigned yet (**type free**) between *free_start_ip_addr* and *free_end_ip_addr*.

ip_id

The database identifier (ID) of the IPv4 address.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

site_name

The name of the space the object belongs to.

tree_level

The database level of the space the object belongs to. If you set up a VLSM organization, it returns values between between 0 (the highest level) and *n*.

tree_path

The database path toward the space the object belongs to as follows: <*space-name*># . If you set up a VLSM organization, the path looks as follows: <*highest-level-space-name*>##<*child-space-name*>#<*child-space-name*>#... .

tree_id_path

The database path toward the space the object belongs to as follows: <*space-ID*># . If you set up a VLSM organization, the path looks as follows: <*highest-level-space-ID*>#<*child-space-ID*>#<*child-space-ID*>#... .

ip_addr

The IPv4 address itself, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr**.

name

The name of the IPv4 address.

mac_addr

The MAC address associated with the IPv4 address.

ip_class_name

The name of the class applied to the IPv4 address, it can be preceded by the class directory.

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. It identifies the parent of the IPv4 network the object belongs to. 0 indicates that the network the object belongs to has no parent network.

parent_subnet_name

The name of the parent IPv4 network:

- # indicates that the network the object belongs to has no parent network.
- *Default* indicates that the network the object belongs to is in an orphan network.

parent_subnet_size

The number of IP addresses of the parent of the network the object belongs to.

parent_vlsm_subnet_id

The database identifier (ID) of the IPv4 subnet-type network, located in the VLSM parent space, from which the parent of the network the IP address belongs to was duplicated. 0 indicates that the parent of the network the IP address belongs to is not a VLSM block-type network duplicated from a parent space.

parent_subnet_class_name

The name of the class applied to the parent of the IPv4 network the object belongs to, it can be preceded by the class directory.

parent_subnet_start_ip_addr

The first IP address of the parent of the IPv4 network the IP address belongs to.

parent_subnet_start_hostaddr

The human readable version of the parameter **parent_subnet_start_ip_addr**.

parent_subnet_end_ip_addr

The last IP address of the parent of the IPv4 network the IP address belongs to.

parent_subnet_end_hostaddr

The human readable version of the parameter **parent_subnet_end_ip_addr**.

subnet_name

The name of the IPv4 network the object belongs to. *Default* indicates that the network the object belongs to is an orphan network.

pool_name

The name of the IPv4 pool the object belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

subnet_id

The database identifier (ID) of the IPv4 network the object belongs to.

subnet_start_ip_addr

The first IP address of the IPv4 network the object belongs to.

subnet_start_hostaddr

The human readable version of the parameter **subnet_start_ip_addr**.

subnet_end_ip_addr

The last IP address of the IPv4 network the object belongs to.

subnet_end_hostaddr

The human readable version of the parameter **subnet_end_ip_addr**.

subnet_size

The number of IP addresses the network the object belongs to contains.

subnet_is_terminal

A way to determine if the network the IP address belongs to is terminal (1) or non-terminal (0).

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network the IP address belongs to.

pool_class_name

The name of the class applied to the IPv4 pool the object belongs to, it can be preceded by the class directory.

pool_id

The database identifier (ID) of the IPv4 pool the object belongs to.

pool_read_only

The reservation status of the pool the IPv4 address belongs to. If set 1, the pool is reserved and you cannot assign the IP address.

pool_row_enabled

Internal use. Not documented.

iplnetdev_name

The name of the NetChange network device associated with the IP address.

iplnetdev_id

The database identifier (ID) of the NetChange network device associated with the IP address.

iplport_name

The name of the NetChange port associated with the IP address.

iplport_slotnumber

The slot number of the port, for IP addresses which MAC addresses is imported from NetChange.

iplport_portnumber

The number of the port, for IP addresses which MAC addresses is imported from NetChange.

iplport_ifvlan

The VLAN identifier (ID) of the NetChange port, for IP addresses which MAC addresses is imported from NetChange.

hostiface_name

The name of the Device Manager interface associated with the IP address.

hostiface_id

The database identifier (ID) of the Device Manager interface associated with the IP address.

hostdev_name

The name of the Device Manager device associated with the IP address.

hostdev_id

The database identifier (ID) of the Device Manager device associated with the IP address.

dhcpghost_id

The database identifier (ID) of the DHCP static associated with the IP address.

dhcplease_id

The database identifier (ID) of the DHCP lease associated with the IP address.

last_seen

The last time the MAC address associated with the IP address was seen on the network, in decimal UNIX date format.

dhcplease_end_time

The expiration time of the lease, if the IP address was imported from the DHCP, in decimal UNIX date format.

site_description

The description of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

subnet_class_name

The name of the class applied to the IPv4 network the object belongs to, it can be preceded by the class directory.

pool_size

The number of IP addresses that contains the pool the IPv4 address belongs to.

pool_start_ip_addr

The first IP address of the IPv4 pool the IP address belongs to.

pool_end_ip_addr

The last IP address of the IPv4 pool the IP address belongs to.

ip_alias

The name of the IPv4 alias(es) associated with the IPv4 address.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 10.3. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|---|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

tag_pool_dhcprange

A way to determine if the pool the IP address belongs to is configured to Create DHCP range (1) or not (0).

tag_container_dhcpstatic

A way to determine if the terminal network or pool the IP address belongs to is configured to Add a DHCP static (1) or not (0).

trace_creation_date

The creation date of the IPv4 address, in decimal UNIX date format.

trace_last_update_date

The last time the IPv4 address was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv4 address.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv4 address.

trace_creation_origin

The name of the module where the IPv4 address addition originated.

trace_creation_exec_stack

The call stack of the IPv4 address operation details, as follows: <service1>&<service2>&<service3>.... .

trace_creation_usr_login

The login of the user who added the IPv4 address.

trace_creation_origin_usr_login

The login of the user who requested the IPv4 address.

ip_class_parameters

The class parameters applied to the IPv4 address and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

ip_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **ip_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

ip_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

pool_class_parameters

The class parameters applied to the IPv4 pool the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

pool_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **pool_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&.... .

subnet_class_parameters

The class parameters applied to the IPv4 network the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet_class_parameters**: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&.... .

Name

ip_used_address_list — List the used IPv4 addresses

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

type

A way to determine if you can assign the IP address (*free*) or if it is In use (*ip*).

free_start_ip_addr

An IP address in hexadecimal format. For addresses *In use* (**type ip**), it returns the IP address itself.

For free addresses (**type free**), it returns the first IP address of a range of IPv4 addresses that are not assigned yet. The last address in that range is returned in *free_end_ip_addr*.

free_end_ip_addr

An IP address in hexadecimal format. For addresses *In use* (**type ip**), it returns the IP address itself.

For free addresses (**type free**), it returns the last IP address of a range of IPv4 addresses that are not assigned yet. The first address in that range is returned in *free_start_ip_addr*.

free_scope_size

The number of IP addresses that are not assigned yet (**type free**) between *free_start_ip_addr* and *free_end_ip_addr*.

ip_id

The database identifier (ID) of the IPv4 address.

site_is_template

The template status of the space the object belongs to. If the space is used as template (1), all the IPv4 networks, pools and IP addresses it contains are also used as template.

site_name

The name of the space the object belongs to.

tree_level

The database level of the space the object belongs to. If you set up a VLSM organization, it returns values between between 0 (the highest level) and *n*.

tree_path

The database path toward the space the object belongs to as follows: <*space-name*># . If you set up a VLSM organization, the path looks as follows: <*highest-level-space-name*>##<*child-space-name*>#<*child-space-name*>#... .

tree_id_path

The database path toward the space the object belongs to as follows: <*space-ID*># . If you set up a VLSM organization, the path looks as follows: <*highest-level-space-ID*>#<*child-space-ID*>#<*child-space-ID*>#... .

ip_addr

The IPv4 address itself, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr**.

name

The name of the IPv4 address.

mac_addr

The MAC address associated with the IPv4 address.

ip_class_name

The name of the class applied to the IPv4 address, it can be preceded by the class directory.

parent_subnet_id

The database identifier (ID) of the parent IPv4 network. It identifies the parent of the IPv4 network the object belongs to. 0 indicates that the network the object belongs to has no parent network.

parent_subnet_name

The name of the parent IPv4 network:

- # indicates that the network the object belongs to has no parent network.
- *Default* indicates that the network the object belongs to is in an orphan network.

parent_subnet_size

The number of IP addresses of the parent of the network the object belongs to.

parent_vlsm_subnet_id

The database identifier (ID) of the IPv4 subnet-type network, located in the VLSM parent space, from which the parent of the network the IP address belongs to was duplicated. 0 indicates that the parent of the network the IP address belongs to is not a VLSM block-type network duplicated from a parent space.

parent_subnet_class_name

The name of the class applied to the parent of the IPv4 network the object belongs to, it can be preceded by the class directory.

parent_subnet_start_ip_addr

The first IP address of the parent of the IPv4 network the IP address belongs to.

parent_subnet_start_hostaddr

The human readable version of the parameter **parent_subnet_start_ip_addr**.

parent_subnet_end_ip_addr

The last IP address of the parent of the IPv4 network the IP address belongs to.

parent_subnet_end_hostaddr

The human readable version of the parameter **parent_subnet_end_ip_addr**.

subnet_name

The name of the IPv4 network the object belongs to. *Default* indicates that the network the object belongs to is an orphan network.

pool_name

The name of the IPv4 pool the object belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

subnet_id

The database identifier (ID) of the IPv4 network the object belongs to.

subnet_start_ip_addr

The first IP address of the IPv4 network the object belongs to.

subnet_start_hostaddr

The human readable version of the parameter **subnet_start_ip_addr**.

subnet_end_ip_addr

The last IP address of the IPv4 network the object belongs to.

subnet_end_hostaddr

The human readable version of the parameter **subnet_end_ip_addr**.

subnet_size

The number of IP addresses the network the object belongs to contains.

subnet_is_terminal

A way to determine if the network the IP address belongs to is terminal (1) or non-terminal (0).

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network the IP address belongs to.

pool_class_name

The name of the class applied to the IPv4 pool the object belongs to, it can be preceded by the class directory.

pool_id

The database identifier (ID) of the IPv4 pool the object belongs to.

pool_read_only

The reservation status of the pool the IPv4 address belongs to. If set 1, the pool is reserved and you cannot assign the IP address.

pool_row_enabled

Internal use. Not documented.

iplnetdev_name

The name of the NetChange network device associated with the IP address.

iplnetdev_id

The database identifier (ID) of the NetChange network device associated with the IP address.

iplport_name

The name of the NetChange port associated with the IP address.

iplport_slotnumber

The slot number of the port, for IP addresses which MAC addresses is imported from NetChange.

iplport_portnumber

The number of the port, for IP addresses which MAC addresses is imported from NetChange.

iplport_ifvlan

The VLAN identifier (ID) of the NetChange port, for IP addresses which MAC addresses is imported from NetChange.

hostiface_name

The name of the Device Manager interface associated with the IP address.

hostiface_id

The database identifier (ID) of the Device Manager interface associated with the IP address.

hostdev_name

The name of the Device Manager device associated with the IP address.

hostdev_id

The database identifier (ID) of the Device Manager device associated with the IP address.

dhcpghost_id

The database identifier (ID) of the DHCP static associated with the IP address.

dhcplease_id

The database identifier (ID) of the DHCP lease associated with the IP address.

last_seen

The last time the MAC address associated with the IP address was seen on the network, in decimal UNIX date format.

dhcplease_end_time

The expiration time of the lease, if the IP address was imported from the DHCP, in decimal UNIX date format.

site_description

The description of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

subnet_class_name

The name of the class applied to the IPv4 network the object belongs to, it can be preceded by the class directory.

pool_size

The number of IP addresses that contains the pool the IPv4 address belongs to.

pool_start_ip_addr

The first IP address of the IPv4 pool the IP address belongs to.

pool_end_ip_addr

The last IP address of the IPv4 pool the IP address belongs to.

ip_alias

The name of the IPv4 alias(es) associated with the IPv4 address.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 10.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|---|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

tag_pool_dhcprange

A way to determine if the pool the IP address belongs to is configured to Create DHCP range (1) or not (0).

tag_container_dhcpstatic

A way to determine if the terminal network or pool the IP address belongs to is configured to Add a DHCP static (1) or not (0).

trace_creation_date

The creation date of the IPv4 address, in decimal UNIX date format.

trace_last_update_date

The last time the IPv4 address was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv4 address.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv4 address.

trace_creation_origin

The name of the module where the IPv4 address addition originated.

trace_creation_exec_stack

The call stack of the IPv4 address operation details, as follows: <service1>&<service2>&<service3>.... .

trace_creation_usr_login

The login of the user who added the IPv4 address.

trace_creation_origin_usr_login

The login of the user who requested the IPv4 address.

ip_class_parameters

The class parameters applied to the IPv4 address and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

ip_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **ip_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

ip_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

pool_class_parameters

The class parameters applied to the IPv4 pool the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

pool_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **pool_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&.... .

subnet_class_parameters

The class parameters applied to the IPv4 network the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet_class_parameters**: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&.... .

Name

ip_address_groupby — Group IPv4 addresses by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: <parameter>='<value>'. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement **SELECT** is returned.

Name

ip_address_groupby_count—Count the number of IPv4 addresses by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

ip_delete — Delete an IPv4 address

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(ip_id || (hostaddr && (site_id || site_name)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

ip_id

The database identifier (ID) of the IPv4 address, a unique numeric key value automatically incremented when you add an IPv4 address. Use the ID to specify the IPv4 address of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

name

The name of the IPv4 address.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ip_name

Deprecated, replaced by **name**.

ip_addr

Deprecated, replaced by **hostaddr**.

hostaddr

The IP address.

| | | | |
|------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
|------|--------------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|----|
| Default value | N/A | Can be edited | No |
|----------------------|-----|----------------------|----|

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 11. IPv6 Address

Name

ip6_address6_add — Add/Edit an IPv6 address

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (hostaddr && (site_id || site_name))
- **Editing:** (ip6_id || (hostaddr && (site_id || site_name)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip6_id

The database identifier (ID) of the IPv6 address, a unique numeric key value automatically incremented when you add an IPv6 address. Use the ID to specify the IPv6 address of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip6_name

The name of the IPv6 address, each IPv6 address must have a unique name.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | | Can be edited | Yes |

ip6_mac_addr

The MAC address you want to associate with the IPv6 address.

| Type | MAC address | Maximum length | 64 |
|------|-------------|----------------|----|
| | | | |

| | | | |
|----------------------|--|----------------------|-----|
| Default value | | Can be edited | Yes |
|----------------------|--|----------------------|-----|

ip6_addr

Deprecated, replaced by **hostaddr**.

hostaddr

The IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

ip6_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

ip6_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

hostdev_id

The database identifier (ID) of the Device Manager device you want to associate with the IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

hostiface_id

The database identifier (ID) of the Device Manager interface you want to associate with the IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

iplport_id

The database identifier (ID) of the NetChange port you want to associate with the IP address.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ip6_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>.class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Example

In the example below, we call the service **ip6_address6_add** with PHP (cURL) to add an IPv6 address in one of our spaces.

Example 11.1. Calling the service ip6_address6_add using PHP

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/ip6_address6_add?" .
                    "hostaddr=2%3A0%3A0%3A0%3A0%3A0%3A6&site_id=44",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "POST",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Name

ip6_address6_list — List the IPv6 addresses

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

type

A way to determine if you can assign the IP address (*free*) or if it is In use (*ip6*).

free_start_ip6_addr

An IP address in hexadecimal format. For addresses *In use* (**type ip6**), it returns the IP address itself.

For free addresses (**type free**), it returns the first IP address of a range of IPv6 addresses that are not assigned yet. The last address in that range is returned in *free_end_ip6_addr*.

free_end_ip6_addr

An IP address in hexadecimal format. For addresses *In use* (**type ip6**), it returns the IP address itself.

For free addresses (**type free**), it returns the last IP address of a range of IPv6 addresses that are not assigned yet. The first address in that range is returned in *free_start_ip6_addr*.

free_scope_size

The number of IP addresses that are not assigned yet (**type free**) between *free_start_ip6_addr* and *free_end_ip6_addr*.

ip6_id

The database identifier (ID) of the IPv6 address.

site_name

The name of the space the object belongs to.

tree_level

The database level of the space the object belongs to. If you set up a VLSM organization, it returns values between between 0 (the highest level) and *n*.

tree_path

The database path toward the space the object belongs to as follows: <space-name># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#... .

parent_subnet6_name

The name of the parent IPv6 network. # indicates that the network the object belongs to has no parent network.

ip6_addr

The IPv6 address itself.

hostaddr

The human readable version of the parameter **ip6_addr**.

ip6_name

The name of the IPv6 address.

ip6_mac_addr

The MAC address associated with the IPv6 address.

ip6_class_name

The name of the class applied to the IPv6 address, it can be preceded by the class directory.

subnet6_name

The name of the IPv6 network the object belongs to.

subnet6_is_terminal

A way to determine if the network the IP address belongs to is terminal (1) or non-terminal (0).

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network the IP address belongs to.

pool6_name

The name of the IPv6 pool the object belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

subnet6_id

The database identifier (ID) of the IPv6 network the object belongs to.

subnet6_start_ip6_addr

The first IP address of the IPv6 network the object belongs to.

subnet6_start_hostaddr

The human readable version of the parameter **subnet6_start_ip6_addr**.

subnet6_end_ip6_addr

The last IP address of the IPv6 network the object belongs to.

subnet6_end_hostaddr

The human readable version of the parameter **subnet6_end_ip6_addr**.

subnet6_size

The number of IP addresses the network the object belongs to contains.

subnet_size

The number of IP addresses the network the object belongs to contains.

subnet6_prefix

The prefix of the IPv6 network the object belongs to.

parent_subnet6_size

The number of IP addresses of the network parent, in hexadecimal format.

parent_subnet6_id

The database identifier (ID) of the parent IPv6 network. It identifies the parent of the IPv6 network the object belongs to. 0 indicates that the network the object belongs to has no parent network.

parent_vlsm_subnet6_id

The database identifier (ID) of the IPv6 subnet-type network, located in the VLSM parent space, from which the parent of the network the IP address belongs to was duplicated. 0 indicates that the parent of the network the IP address belongs to is not a VLSM block-type network duplicated from a parent space.

pool6_class_name

The name of the class applied to the IPv6 pool the object belongs to, it can be preceded by the class directory.

pool6_id

The database identifier (ID) of the IPv6 pool the object belongs to.

pool6_read_only

The reservation status of the pool the IPv6 address belongs to. If set 1, the pool is reserved and you cannot assign the IP address.

pool6_row_enabled

Internal use. Not documented.

hostiface_name

The name of the Device Manager interface associated with the IP address.

hostiface_id

The database identifier (ID) of the Device Manager interface associated with the IP address.

hostdev_name

The name of the Device Manager device associated with the IP address.

hostdev_id

The database identifier (ID) of the Device Manager device associated with the IP address.

site_description

The description of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

parent_site_name

The name of the space where is located the parent of the network the IPv6 address belongs to. # indicates that the network the IPv6 address belongs to has no parent network.

parent_subnet6_class_name

The name of the class applied to the parent of the IPv6 network the object belongs to, it can be preceded by the class directory.

parent_subnet6_prefix

The prefix of the parent of the IPv6 network the object belongs to.

parent_subnet6_start_ip6_addr

The first IP address of the parent of the IPv6 network the IP address belongs to.

parent_subnet6_start_hostaddr

The human readable version of the parameter **parent_subnet6_start_ip6_addr**.

parent_subnet6_end_ip6_addr

The last IP address of the parent of the IPv6 network the IP address belongs to.

parent_subnet6_end_hostaddr

The human readable version of the parameter **parent_subnet6_end_ip6_addr**.

vlsm_subnet6_id

The database identifier (ID) of the IPv6 subnet-type network, located in the VLSM parent space, from which the network the IP address belongs to was duplicated. 0 indicates the network the IP address belongs to is not a VLSM block-type network duplicated from a parent space.

subnet6_class_name

The name of the class applied to the IPv6 network the object belongs to, it can be preceded by the class directory.

pool6_size

The number of IP addresses that contains the pool the IPv6 address belongs to.

pool6_start_ip6_addr

The first IP address of the IPv6 pool the IP address belongs to.

pool6_end_ip6_addr

The last IP address of the IPv6 pool the IP address belongs to.

ip6_alias

The name of the IPv6 alias(es) associated with the IPv6 address.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 11.1. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

trace_creation_date

The creation date of the IPv6 address, in decimal UNIX date format.

trace_last_update_date

The last time the IPv6 address was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv6 address.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv6 address.

trace_creation_origin

The name of the module where the IPv6 address addition originated.

trace_creation_exec_stack

The call stack of the IPv6 address operation details, as follows: <service1>&<service2>&<service3>....

trace_creation_usr_login

The login of the user who added the IPv6 address.

trace_creation_origin_usr_login

The login of the user who requested the IPv6 address.

ip6_class_parameters

The class parameters applied to the IPv6 address and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

ip6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **ip6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

ip6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

pool6_class_parameters

The class parameters applied to the IPv6 pool the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

pool6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **pool6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&.... .

subnet6_class_parameters

The class parameters applied to the IPv6 network the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

ip6_address6_info — Display the properties of an IPv6 address

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

ip6_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

ip6_id

The database identifier (ID) of the IPv6 address, a unique numeric key value automatically incremented when you add an IPv6 address. Use the ID to specify the IPv6 address of your choice.

Output Parameters

type

A way to determine if you can assign the IP address (`free`) or if it is In use (`ip6`).

free_start_ip6_addr

An IP address in hexadecimal format. For addresses `In use` (`type ip6`), it returns the IP address itself.

For free addresses (`type free`), it returns the first IP address of a range of IPv6 addresses that are not assigned yet. The last address in that range is returned in `free_end_ip6_addr`.

free_end_ip6_addr

An IP address in hexadecimal format. For addresses `In use` (`type ip6`), it returns the IP address itself.

For free addresses (**type free**), it returns the last IP address of a range of IPv6 addresses that are not assigned yet. The first address in that range is returned in *free_start_ip6_addr*.

free_scope_size

The number of IP addresses that are not assigned yet (**type free**) between *free_start_ip6_addr* and *free_end_ip6_addr*.

ip6_id

The database identifier (ID) of the IPv6 address.

site_name

The name of the space the object belongs to.

tree_level

The database level of the space the object belongs to. If you set up a VLSM organization, it returns values between 0 (the highest level) and *n*.

tree_path

The database path toward the space the object belongs to as follows: <space-name># . If you set up a VLSM organization, the path looks as follows: <highest-level-space-name>##<child-space-name>#<child-space-name>#... .

parent_subnet6_name

The name of the parent IPv6 network. # indicates that the network the object belongs to has no parent network.

ip6_addr

The IPv6 address itself.

hostaddr

The human readable version of the parameter **ip6_addr**.

ip6_name

The name of the IPv6 address.

ip6_mac_addr

The MAC address associated with the IPv6 address.

ip6_class_name

The name of the class applied to the IPv6 address, it can be preceded by the class directory.

subnet6_name

The name of the IPv6 network the object belongs to.

subnet6_is_terminal

A way to determine if the network the IP address belongs to is terminal (1) or non-terminal (0).

lock_network_broadcast

A way to prevent (1) users from assigning the broadcast IP address and network IP address of the network the IP address belongs to.

pool6_name

The name of the IPv6 pool the object belongs to.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

subnet6_id

The database identifier (ID) of the IPv6 network the object belongs to.

subnet6_start_ip6_addr

The first IP address of the IPv6 network the object belongs to.

subnet6_start_hostaddr

The human readable version of the parameter **subnet6_start_ip6_addr**.

subnet6_end_ip6_addr

The last IP address of the IPv6 network the object belongs to.

subnet6_end_hostaddr

The human readable version of the parameter **subnet6_end_ip6_addr**.

subnet6_size

The number of IP addresses the network the object belongs to contains.

subnet_size

The number of IP addresses the network the object belongs to contains.

subnet6_prefix

The prefix of the IPv6 network the object belongs to.

parent_subnet6_size

The number of IP addresses of the network parent, in hexadecimal format.

parent_subnet6_id

The database identifier (ID) of the parent IPv6 network. It identifies the parent of the IPv6 network the object belongs to. *0* indicates that the network the object belongs to has no parent network.

parent_vlsm_subnet6_id

The database identifier (ID) of the IPv6 subnet-type network, located in the VLSM parent space, from which the parent of the network the IP address belongs to was duplicated. *0* indicates that the parent of the network the IP address belongs to is not a VLSM block-type network duplicated from a parent space.

pool6_class_name

The name of the class applied to the IPv6 pool the object belongs to, it can be preceded by the class directory.

pool6_id

The database identifier (ID) of the IPv6 pool the object belongs to.

pool6_read_only

The reservation status of the pool the IPv6 address belongs to. If set *1*, the pool is reserved and you cannot assign the IP address.

pool6_row_enabled

Internal use. Not documented.

hostiface_name

The name of the Device Manager interface associated with the IP address.

hostiface_id

The database identifier (ID) of the Device Manager interface associated with the IP address.

hostdev_name

The name of the Device Manager device associated with the IP address.

hostdev_id

The database identifier (ID) of the Device Manager device associated with the IP address.

site_description

The description of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

parent_site_name

The name of the space where is located the parent of the network the IPv6 address belongs to. # indicates that the network the IPv6 address belongs to has no parent network.

parent_subnet6_class_name

The name of the class applied to the parent of the IPv6 network the object belongs to, it can be preceded by the class directory.

parent_subnet6_prefix

The prefix of the parent of the IPv6 network the object belongs to.

parent_subnet6_start_ip6_addr

The first IP address of the parent of the IPv6 network the IP address belongs to.

parent_subnet6_start_hostaddr

The human readable version of the parameter **parent_subnet6_start_ip6_addr**.

parent_subnet6_end_ip6_addr

The last IP address of the parent of the IPv6 network the IP address belongs to.

parent_subnet6_end_hostaddr

The human readable version of the parameter **parent_subnet6_end_ip6_addr**.

vlsm_subnet6_id

The database identifier (ID) of the IPv6 subnet-type network, located in the VLSM parent space, from which the network the IP address belongs to was duplicated. 0 indicates the network the IP address belongs to is not a VLSM block-type network duplicated from a parent space.

subnet6_class_name

The name of the class applied to the IPv6 network the object belongs to, it can be preceded by the class directory.

pool6_size

The number of IP addresses that contains the pool the IPv6 address belongs to.

pool6_start_ip6_addr

The first IP address of the IPv6 pool the IP address belongs to.

pool6_end_ip6_addr

The last IP address of the IPv6 pool the IP address belongs to.

ip6_alias

The name of the IPv6 alias(es) associated with the IPv6 address.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 11.2. Multi-status severity levels

| Message number | Severity | Description |
|-----------------------|-----------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

trace_creation_date

The creation date of the IPv6 address, in decimal UNIX date format.

trace_last_update_date

The last time the IPv6 address was updated, in decimal UNIX date format.

trace_creation_usr_id

The database identifier (ID) of the user who added the IPv6 address.

trace_creation_origin_usr_id

The database identifier (ID) of the user who requested the IPv6 address.

trace_creation_origin

The name of the module where the IPv6 address addition originated.

trace_creation_exec_stack

The call stack of the IPv6 address operation details, as follows: <service1>&<service2>&<service3>... .

trace_creation_usr_login

The login of the user who added the IPv6 address.

trace_creation_origin_usr_login

The login of the user who requested the IPv6 address.

ip6_class_parameters

The class parameters applied to the IPv6 address and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

ip6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **ip6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

ip6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

pool6_class_parameters

The class parameters applied to the IPv6 pool the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

pool6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **pool6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

If the inheritance or propagation property is not specified, its default value - *set, propagate* - is used.

site_class_parameters

The class parameters applied to the space the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

site_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **site_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>,<propagation>&.... .

subnet6_class_parameters

The class parameters applied to the IPv6 network the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

subnet6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **subnet6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

ip6_address6_count — Count the number of IPv6 addresses

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

ip6_find_free_address6 — List the free IPv6 addresses

Description

This service allows you to list the 10 first free IPv6 addresses.

You must execute the service using **rpc**.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(subnet6_id || pool6_id || parent_subnet6_id)

Input Parameters

subnet6_id

The database identifier (ID) of the IPv6 network, a unique numeric key value automatically incremented when you add an IPv6 network. Use the ID to specify the IPv6 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

parent_subnet6_id

The database identifier (ID) of the parent IPv6 network. Use the ID to specify the parent IPv6 network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

pool6_id

The database identifier (ID) of the IPv6 pool, a unique numeric key value automatically incremented when you add an IPv6 pool. Use the ID to specify the IPv6 pool of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

max_find

The maximum number of IPv6 addresses to be returned by the service. You can use it to return more than 10 results.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

| | | | |
|---------------|-----|----------------|-----|
| Type | | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

begin_addr

The first IPv6 address of the range of addresses where you are looking for free IP addresses.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

The last IPv6 address of the range of addresses where you are looking for free IP addresses.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

pool6_class_name

The name of the class applied to the IPv6 pool the IP addresses you are looking for belong to. You must specify the class file directory, e.g. *my_directory/my_class.class* . You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

subnet6_class_name

The name of the class applied to the IPv6 network the IP addresses you are looking for belong to. You must specify the class file directory, e.g. *my_directory/my_class.class* . You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

ip6_addr

The IPv6 address itself.

hostaddr6

The IP address.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

site_name

The name of the space the object belongs to.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

subnet_id

The database identifier (ID) of the IPv6 network the object belongs to.

subnet6_name

The name of the IPv6 network the object belongs to.

pool6_id

The database identifier (ID) of the IPv6 pool the object belongs to.

pool6_name

The name of the IPv6 pool the object belongs to.

Name

ip6_address6_groupby — Group IPv6 addresses by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

ip6_address6_groupby_count — Count the number of IPv6 addresses by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

ip6_address6_delete — Delete an IPv6 address

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(ip6_id || (hostaddr && (site_id || site_name)))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

ip6_id

The database identifier (ID) of the IPv6 address, a unique numeric key value automatically incremented when you add an IPv6 address. Use the ID to specify the IPv6 address of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ip6_name

The name of the IPv6 address.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ip6_addr

Deprecated, replaced by **hostaddr**.

hostaddr

The IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Example

In the example below, we call the service **ip6_address6_delete** with Ruby (NET::Http) to delete a specific IPv6 address.

Example 11.2. Calling the service ip6_address6_delete using Ruby

```
require 'uri'
require 'net/http'

url = URI("https://solid.intranet/rest/ip6_address6_delete?ip6_id=17")

http = Net::HTTP.new(url.host, url.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE

request = Net::HTTP::Delete.new(url)
request["x-ipm-username"] = 'aXBtYWRtaW4='
request["x-ipm-password"] = 'YWRtaW4='
request["cache-control"] = 'no-cache'

response = http.request(request)
puts response.read_body
```

Chapter 12. IPv4 Address Alias

Name

ip_alias_add — Add/Edit an IPv4 address alias

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (ip_name && (ip_id || (hostaddr && (site_id || site_name))))
- **Editing:** (ip_name_id || (ip_name && (ip_id || (hostaddr && (site_id || site_name)))))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip_id

The database identifier (ID) of the IPv4 address, a unique numeric key value automatically incremented when you add an IPv4 address. Use the ID to specify the IPv4 address of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip_name_id

The database identifier (ID) of the IPv4 alias, a unique numeric key value automatically incremented when you add an IPv4 alias. Use the ID to specify the IPv4 alias of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip_name

The name of the IPv4 alias.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

name

Deprecated, replaced by **ip_name**.

ip_name_type

The type of the alias.

| | | | |
|---------------|---------------------------------------|----------------|-----|
| Type | Fixed value: A a CNAME cname | Maximum length | N/A |
| Default value | CNAME | Can be edited | No |

ip_addr

Deprecated, replaced by **hostaddr**.

hostaddr

The IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

ip_alias_list — List the aliases of an IPv4 address

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

ip_id

The database identifier (ID) of the IPv4 address, a unique numeric key value automatically incremented when you add an IPv4 address. Use the ID to specify the IPv4 address of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

alias_name

The name of the alias.

ip_name_type

The type of the alias, either **CNAME** or **A**.

ip_name_id

The database identifier (ID) of the IPv4 alias.

ip_id

The database identifier (ID) of the IPv4 address associated with the alias.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

ip_addr

The IPv4 alias itself, in hexadecimal format.

name

The name of the IPv4 alias.

mac_addr

The MAC address associated with the IPv4 alias.

ip_class_name

The name of the class applied to the object, it can be preceded by the class directory.

subnet_id

The database identifier (ID) of the IPv4 network the object belongs to.

ip_type

A way to determine if you can assign the IP alias (**free**) or if it is In use (**ip**).

pool_id

The database identifier (ID) of the IPv4 pool the object belongs to.

iplport_id

The database identifier (ID) of the NetChange port associated with the IP address.

hostdev_id

The database identifier (ID) of the Device Manager device associated with the IP address.

hostiface_id

The database identifier (ID) of the Device Manager interface associated with the IP address.

dhcpghost_id

The database identifier (ID) of the DHCP static associated with the IP address.

dhcplease_id

The database identifier (ID) of the DHCP lease associated with the IP address.

site_name

The name of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

ip_class_parameters

The class parameters applied to the IPv4 alias and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

ip_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by

ip_class_parameters: <class-parameter1>=<inheritance>, <propagation>&<class-parameter2>=<inheritance>&.... .

ip_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:

<class-parameter1>=real_<container-type>, <container-ID>&<class-parameter2>=real_<container-type>, <container-ID>&.... .

Example

In the example below, we call the service **ip_alias_list** with Python (Requests) using the clause **ORDERBY** to list the 5 aliases of an IPv4 address and sort them based on their name and type.

Example 12.1. Calling the service ip_alias_list using Python and ORDERBY

```
import requests

url = "https://solid.intranet/rest/ip_alias_list"

querystring = {"ip_id": "200", "limit": "10", "ORDERBY": "alias_name, ip_name_type"}

headers = {
    'x-ipm-username': "aXBtYWRtaW4=",
    'x-ipm-password': "YWRTaw4=",
    'cache-control': "no-cache"
}

response = requests.request("GET", url, headers=headers, params=querystring)

print(response.text)
```

Name

ip_alias_count — Count the number of aliases of an IPv4 address

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

ip_id

The database identifier (ID) of the IPv4 address, a unique numeric key value automatically incremented when you add an IPv4 address. Use the ID to specify the IPv4 address of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

ip_alias_delete — Delete an IPv4 address alias

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(ip_name_id || (ip_name && (ip_id || (hostaddr && (site_id || site_name))))))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip_id

The database identifier (ID) of the IPv4 address, a unique numeric key value automatically incremented when you add an IPv4 address. Use the ID to specify the IPv4 address of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip_name_id

The database identifier (ID) of the IPv4 alias, a unique numeric key value automatically incremented when you add an IPv4 alias. Use the ID to specify the IPv4 alias to of your choice.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip_name

The name of the IPv4 alias.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip_name_type

The type of the alias.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ip_addr

Deprecated, replaced by **hostaddr**.

hostaddr

The IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 13. IPv6 Address Alias

Name

ip6_alias_add — Add/Edit an IPv6 address alias

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (ip6_name && (ip6_id || (hostaddr && (site_id || site_name))))
- **Editing:** (ip6_name_id || (ip6_name && (ip6_id || (hostaddr && (site_id || site_name)))))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip6_id

The database identifier (ID) of the IPv6 address, a unique numeric key value automatically incremented when you add an IPv6 address. Use the ID to specify the IPv6 address of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip6_name_id

The database identifier (ID) of the IPv6 alias, a unique numeric key value automatically incremented when you add an IPv6 alias. Use the ID to specify the IPv6 alias of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip6_name

The name of the IPv6 address.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

name

Deprecated, replaced by **ip6_name**.

ip6_name_type

The type of the alias.

| | | | |
|---------------|---|----------------|-----|
| Type | Fixed value: AAAA aaaa CNAME cname | Maximum length | N/A |
| Default value | CNAME | Can be edited | No |

ip6_addr

Deprecated, replaced by **hostaddr**.

hostaddr

The IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

ip6_alias_list — List the aliases of an IPv6 address

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

ip6_id

The database identifier (ID) of the IPv6 address, a unique numeric key value automatically incremented when you add an IPv6 address. Use the ID to specify the IPv6 address of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

alias_name

The name of the alias.

ip6_name_type

The type of the alias, either **CNAME** or **AAAA**.

ip6_name_id

The database identifier (ID) of the IPv6 alias.

ip6_id

The database identifier (ID) of the IPv6 address associated with the alias.

site_id

The database identifier (ID) of the space the object belongs to, a unique numeric key value automatically incremented when you add a space.

ip6_addr

The IPv6 alias itself.

ip6_name

The name of the IPv6 address associated with the alias.

ip6_mac_addr

The MAC address associated with the IPv6 alias.

subnet6_id

The database identifier (ID) of the IPv6 network the object belongs to.

pool6_id

The database identifier (ID) of the IPv6 pool the object belongs to.

iplport_id

The database identifier (ID) of the NetChange port associated with the IP address.

hostdev_id

The database identifier (ID) of the Device Manager device associated with the IP address.

hostiface_id

The database identifier (ID) of the Device Manager interface associated with the IP address.

ip6_class_name

The name of the class applied to the object, it can be preceded by the class directory.

site_name

The name of the space the object belongs to.

site_class_name

The name of the class applied to the space the object belongs to, it can be preceded by the class directory.

ip6_class_parameters

The class parameters applied to the IPv6 alias and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

ip6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
ip6_class_parameters: <class-parameter1>=<inheritance>, <propagation>&<class-parameter2>=<inheritance>&.... .

ip6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:
<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

ip6_alias_count — Count the number of aliases of an IPv6 address

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

ip6_id

The database identifier (ID) of the IPv6 address, a unique numeric key value automatically incremented when you add an IPv6 address. Use the ID to specify the IPv6 address of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

ip6_alias_delete — Delete an IPv6 address alias

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(ip6_name_id || (ip6_name && (ip6_id || (hostaddr && (site_id || site_name)))))

Input Parameters

site_id

The database identifier (ID) of the space, a unique numeric key value automatically incremented when you add a space. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip6_id

The database identifier (ID) of the IPv6 address, a unique numeric key value automatically incremented when you add an IPv6 address. Use the ID to specify the IPv6 address of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip6_name_id

The database identifier (ID) of the IPv6 alias, a unique numeric key value automatically incremented when you add an IPv6 alias. Use the ID to specify the IPv6 alias to of your choice.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip6_name

The name of the IPv6 address.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ip6_name_type

The type of the alias.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ip6_addr

Deprecated, replaced by **hostaddr**.

hostaddr

The IP address.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Part III. DHCP Services

Table of Contents

| | |
|-----------------------------------|-----|
| 14. DHCPv4 Server | 258 |
| dhcp_server_list | 259 |
| dhcp_server_info | 265 |
| dhcp_server_count | 271 |
| dhcp_server_options_list | 272 |
| 15. DHCPv6 Server | 275 |
| dhcp6_server6_list | 276 |
| dhcp6_server6_info | 281 |
| dhcp6_server6_count | 286 |
| dhcp6_server6_options6_list | 287 |
| 16. DHCPv4 Shared Network | 290 |
| dhcp_sn_add | 291 |
| dhcp_shared_network_list | 293 |
| dhcp_shared_network_info | 296 |
| dhcp_shared_network_count | 298 |
| dhcp_sn_delete | 299 |
| 17. DHCPv6 Shared Network | 301 |
| dhcp6_sn6_add | 302 |
| dhcp6_shared_network6_list | 304 |
| dhcp6_shared_network6_info | 307 |
| dhcp6_shared_network6_count | 309 |
| dhcp6_sn6_delete | 310 |
| 18. DHCPv4 Scope | 312 |
| dhcp_scope_add | 313 |
| dhcp_scope_list | 318 |
| dhcp_scope_info | 323 |
| dhcp_scope_count | 327 |
| dhcp_scope_groupby | 328 |
| dhcp_scope_groupby_count | 330 |
| dhcp_scope_options_list | 332 |
| group_dhcpscope_add | 335 |
| group_dhcpscope_delete | 338 |
| dhcp_scope_delete | 340 |
| 19. DHCPv6 Scope | 342 |
| dhcp6_scope6_add | 343 |
| dhcp6_scope6_list | 347 |
| dhcp6_scope6_info | 351 |
| dhcp6_scope6_count | 355 |
| dhcp6_scope6_options6_list | 356 |
| group_dhcpscope6_add | 359 |
| group_dhcpscope6_delete | 361 |
| dhcp6_scope6_delete | 363 |
| 20. DHCPv4 Group | 365 |
| dhcp_group_add | 366 |
| dhcp_group_list | 369 |
| dhcp_group_info | 373 |
| dhcp_group_count | 376 |
| dhcp_group_delete | 377 |
| 21. DHCPv6 Group | 379 |
| dhcp6_group6_list | 380 |
| 22. DHCPv4 Range | 383 |

| | |
|--------------------------------------|-----|
| dhcp_range_add | 384 |
| dhcp_range_list | 388 |
| dhcp_range_info | 393 |
| dhcp_range_count | 397 |
| dhcp_range_options_list | 398 |
| dhcp_range_delete | 401 |
| 23. DHCPv6 Range | 404 |
| dhcp6_range6_add | 405 |
| dhcp6_range6_list | 409 |
| dhcp6_range6_info | 414 |
| dhcp6_range6_count | 418 |
| dhcp6_range6_options6_list | 419 |
| dhcp6_range6_delete | 422 |
| 24. DHCPv4 Lease | 424 |
| dhcp_range_lease_list | 425 |
| dhcp_range_lease_info | 430 |
| dhcp_range_lease_count | 434 |
| dhcp_range_lease_groupby | 435 |
| dhcp_range_lease_groupby_count | 437 |
| dhcp_lease_log_list | 439 |
| dhcp_lease_log_count | 442 |
| dhcp_lease_log_groupby | 443 |
| dhcp_lease_log_groupby_count | 445 |
| dhcp_lease_manual_delete | 447 |
| 25. DHCPv6 Lease | 449 |
| dhcp6_lease6_list | 450 |
| dhcp6_lease6_count | 454 |
| dhcp6_lease6_log_list | 455 |
| dhcp6_lease6_log_count | 458 |
| dhcp6_lease6_log_groupby | 459 |
| dhcp6_lease6_log_groupby_count | 461 |
| 26. DHCPv4 Static | 463 |
| dhcp_static_add | 464 |
| dhcp_static_list | 469 |
| dhcp_static_info | 474 |
| dhcp_static_count | 479 |
| dhcp_static_groupby | 480 |
| dhcp_static_groupby_count | 482 |
| dhcp_static_options_list | 484 |
| dhcp_static_delete | 487 |
| 27. DHCPv6 Static | 490 |
| dhcp6_static6_add | 491 |
| dhcp6_static6_list | 495 |
| dhcp6_static6_info | 500 |
| dhcp6_static6_count | 504 |
| dhcp6_static6_options6_list | 505 |
| dhcp6_static6_delete | 508 |
| 28. DHCPv4 Option | 510 |
| dhcp_option_add | 511 |
| 29. DHCPv6 Option | 516 |
| dhcp6_option6_add | 517 |
| 30. DHCPv4 ACL and ACL Entry | 521 |
| dhcp_acl_add | 522 |
| dhcp_class_list | 525 |

| | |
|--------------------------------------|-----|
| dhcp_class_info | 527 |
| dhcp_class_count | 529 |
| dhcp_acl_delete | 530 |
| dhcp_acl_data_add | 532 |
| dhcp_subclass_list | 535 |
| dhcp_subclass_info | 538 |
| dhcp_subclass_count | 540 |
| dhcp_acl_data_delete | 541 |
| 31. DHCPv6 ACL and ACL Entry | 543 |
| dhcp6_acl6_add | 544 |
| dhcp6_class6_list | 547 |
| dhcp6_class6_info | 549 |
| dhcp6_class6_count | 551 |
| dhcp6_acl6_delete | 552 |
| dhcp6_acl_data6_add | 554 |
| dhcp6_subclass6_list | 557 |
| dhcp6_subclass6_info | 560 |
| dhcp6_subclass6_count | 562 |
| dhcp6_acl_data6_delete | 563 |
| 32. DHCPv4 Failover Channel | 565 |
| dhcp_failover_list | 566 |
| dhcp_failover_info | 569 |
| dhcp_failover_count | 572 |
| dhcp_failover_set_partner_down | 573 |

Chapter 14. DHCPv4 Server

Name

dhcp_server_list — List the DHCPv4 servers

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcp_localtime

The local time on the DHCPv4 server, in decimal UNIX date format.

msrpc_login

The login of the Microsoft Windows DHCP server.

msrpc_domain

The domain name of the Microsoft Windows DHCP server.

ipmdhcp_protocol

Internal use. Not documented.

ipmdhcp_https_login

Internal use. Not documented.

connectionprofile_name

The name of the connection profile used as connection method for the DHCPv4 server.

ipmdhcp_is_package

The DHCPv4 server package information. *Y* for an EfficientIP Package server, *N* for an appliance or virtual machine, *U* the package information is irrelevant. For servers with a **dhcp_type** set to *ipm*, *U* indicates either EfficientIP Packages or appliances/virtual machines.

isolated

A way to determine if the server can update any other module (1).

tree_path

The database path toward the server as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

vdhcp_param1

Internal use. Not documented.

tcp_port

Internal use. Not documented.

ms_use_ssl

Internal use. Not documented.

windhcp_protocol

Internal use. Not documented.

snmp_id

Internal use. Not documented.

snmp_port

Internal use. Not documented.

snmp_profile_id

Internal use. Not documented.

snmp_retry

Internal use. Not documented.

snmp_timeout

Internal use. Not documented.

snmp_use_tcp

Internal use. Not documented.

cisco_use_ssh

Internal use. Not documented.

cisco_login

Internal use. Not documented.

cisco_password

Internal use. Not documented.

cisco_root_password

Internal use. Not documented.

ref1_dhcp_name

The name of the Master or Single DHCPv4 server within the smart architecture.

vdhcp_ref1_dhcp_id

The database identifier (ID) of the DHCPv4 smart architecture the server belongs to.

ref2_dhcp_name

Internal use. Not documented.

vdhcp_ref2_dhcp_id

Internal use. Not documented.

tree_level

The database level of the server. 0 indicates the server is managed on its own, 1 indicates it is managed by a smart architecture.

total_vdhcp_members

The total number of servers managed by the DHCPv4 smart architecture.

vdhcp_members_name

The list of the servers managed by the DHCPv4 smart architecture, as follows: <dhcp_name>,<dhcp_name>,... .

vdhcp_arch

The type of the DHCPv4 smart architecture:

Table 14.1. vdhcp_arch possible values

| Type | Description |
|-------------|--|
| masterslave | The One-to-One smart architecture sets a pair of DHCP servers in a Master/Backup configuration. |
| star | The One-to-Many smart architecture sets a multi-site failover configuration at the cost of n-servers+1. |
| splitscope | The Split-Scope smart architecture sets a pair of DHCP servers in a configuration where the two scopes listen to the same subnet, but the range of addresses is divided. |
| single | The Single-Server smart architecture manages a single DHCP server. |

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp_parent_arch

The type of the DHCPv4 smart architecture managing the DHCPv4 server. No value indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server. 0 indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp_ref1_dhcp_name

Internal use. Not documented.

vdhcp_ref2_dhcp_name

Internal use. Not documented.

dhcp_uboottime

Internal use. Not documented.

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server.

dhcp_type

The type of the DHCPv4 server:

Table 14.2. dhcp_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

dhcp_state

The status of the DHCPv4 server:

Table 14.3. dhcp_state possible values

| Status | Description |
|--------|---|
| ER | The license used in SOLIDserver is not compliant with the added server: the license is invalid. |
| ES | The server configuration could not be parsed properly. |
| ET | The server does not answer anymore due to a scheduled configuration of the server. |
| IS | There was a setting error during the server declaration. For instance, some settings were added to a server that does not support them or a smart architecture is not managing any physical server. |
| IC | The SSL credentials are invalid |
| IP | The account used to add the Microsoft Windows DHCP server does not have sufficient privileges to manage it. |
| LS | The server configuration is not viable. |
| N | The server does not have a status as it has not synchronized yet. |
| Y | The server is operational. |

dhcp_synching

The synchronization status of the DHCPv4 server. 1 indicates that the server is currently being synchronized.

dhcp_name

The name of the DHCPv4 server.

dhcp_comment

The description of the DHCPv4 server.

dhcp_version

The version details of the DHCPv4 server.

row_enabled

The object activation status:

- If set to *0*, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to *1*, the object is enabled and managed.

By default, *row_enabled* is set to *1* when an object is created.

dhcp_class_name

The name of the class applied to the DHCPv4 server, it can be preceded by the class directory.

ip_addr

The Management IP address of the DHCPv4 server, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

service_ip_addr

The Service IP address of the DHCPv4 server, the IPv4 address configured when adding the server.

stat_enabled

Internal use. Not documented.

stat_period

Internal use. Not documented.

stat_niceness

Internal use. Not documented.

stat_time

Internal use. Not documented.

reverse_proxy_conf

The URL of the HTTP(S) reverse proxy server that forwards client requests to the DHCPv4 server, if you configured one.

cluster_role

The role of the server in the cluster, either *active (M)*, *passive (B)* or *N/A (#)*.

cluster_peer_dhcp_id

The database identifier (ID) of the other server of the cluster.

cluster_hb_hostaddr

The IP address of the heartbeat of the server, a direct link between the servers of the cluster.

cluster_ssh_keyring_id

The database identifier (ID) of the SSH key dedicated to the cluster communication.

cluster_vip_phys_hostaddr

The local physical IP address of the VIP the cluster relies on.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 14.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcp_class_parameters

The class parameters applied to the DHCPv4 server and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcp_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

dhcp_server_info — Display the properties of a DHCPv4 server

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcp_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

Output Parameters

dhcp_localtime

The local time on the DHCPv4 server, in decimal UNIX date format.

msrpc_login

The login of the Microsoft Windows DHCP server.

msrpc_domain

The domain name of the Microsoft Windows DHCP server.

ipmdhcp_protocol

Internal use. Not documented.

ipmdhcp_https_login

Internal use. Not documented.

connectionprofile_name

The name of the connection profile used as connection method for the DHCPv4 server.

ipmdhcp_is_package

The DHCPv4 server package information. *Y* for an EfficientIP Package server, *N* for an appliance or virtual machine, *U* the package information is irrelevant. For servers with a **dhcp_type** set to *ipm*, *U* indicates either EfficientIP Packages or appliances/virtual machines.

isolated

A way to determine if the server can update any other module (1).

tree_path

The database path toward the server as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

vdhcp_param1

Internal use. Not documented.

tcp_port

Internal use. Not documented.

ms_use_ssl

Internal use. Not documented.

windhcp_protocol

Internal use. Not documented.

snmp_id

Internal use. Not documented.

snmp_port

Internal use. Not documented.

snmp_profile_id

Internal use. Not documented.

snmp_retry

Internal use. Not documented.

snmp_timeout

Internal use. Not documented.

snmp_use_tcp

Internal use. Not documented.

cisco_use_ssh

Internal use. Not documented.

cisco_login

Internal use. Not documented.

cisco_password

Internal use. Not documented.

cisco_root_password

Internal use. Not documented.

ref1_dhcp_name

The name of the Master or Single DHCPv4 server within the smart architecture.

vdhcp_ref1_dhcp_id

The database identifier (ID) of the DHCPv4 smart architecture the server belongs to.

ref2_dhcp_name

Internal use. Not documented.

vdhcp_ref2_dhcp_id

Internal use. Not documented.

tree_level

The database level of the server. *0* indicates the server is managed on its own, *1* indicates it is managed by a smart architecture.

total_vdhcp_members

The total number of servers managed by the DHCPv4 smart architecture.

vdhcp_members_name

The list of the servers managed by the DHCPv4 smart architecture, as follows: <dhcp_name>,<dhcp_name>,... .

vdhcp_arch

The type of the DHCPv4 smart architecture:

Table 14.5. vdhcp_arch possible values

| Type | Description |
|-------------|--|
| masterslave | The One-to-One smart architecture sets a pair of DHCP servers in a Master/Backup configuration. |
| star | The One-to-Many smart architecture sets a multi-site failover configuration at the cost of n-servers+1. |
| splitscope | The Split-Scope smart architecture sets a pair of DHCP servers in a configuration where the two scopes listen to the same subnet, but the range of addresses is divided. |
| single | The Single-Server smart architecture manages a single DHCP server. |

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp_parent_arch

The type of the DHCPv4 smart architecture managing the DHCPv4 server. No value indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server. *0* indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp_ref1_dhcp_name

Internal use. Not documented.

vdhcp_ref2_dhcp_name

Internal use. Not documented.

dhcp_uboottime

Internal use. Not documented.

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server.

dhcp_type

The type of the DHCPv4 server:

Table 14.6. dhcp_type possible values

| Type | Description |
|------|---|
| ipm | EfficientIP or EfficientIP Package server |

| Type | Description |
|-------|---------------------------------------|
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

dhcp_state

The status of the DHCPv4 server:

Table 14.7. *dhcp_state* possible values

| Status | Description |
|--------|---|
| ER | The license used in SOLIDserver is not compliant with the added server: the license is invalid. |
| ES | The server configuration could not be parsed properly. |
| ET | The server does not answer anymore due to a scheduled configuration of the server. |
| IS | There was a setting error during the server declaration. For instance, some settings were added to a server that does not support them or a smart architecture is not managing any physical server. |
| IC | The SSL credentials are invalid |
| IP | The account used to add the Microsoft Windows DHCP server does not have sufficient privileges to manage it. |
| LS | The server configuration is not viable. |
| N | The server does not have a status as it has not synchronized yet. |
| Y | The server is operational. |

dhcp_synching

The synchronization status of the DHCPv4 server. 1 indicates that the server is currently being synchronized.

dhcp_name

The name of the DHCPv4 server.

dhcp_comment

The description of the DHCPv4 server.

dhcp_version

The version details of the DHCPv4 server.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

dhcp_class_name

The name of the class applied to the DHCPv4 server, it can be preceded by the class directory.

ip_addr

The Management IP address of the DHCPv4 server, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

service_ip_addr

The Service IP address of the DHCPv4 server, the IPv4 address configured when adding the server.

stat_enabled

Internal use. Not documented.

stat_period

Internal use. Not documented.

stat_niceness

Internal use. Not documented.

stat_time

Internal use. Not documented.

reverse_proxy_conf

The URL of the HTTP(S) reverse proxy server that forwards client requests to the DHCPv4 server, if you configured one.

cluster_role

The role of the server in the cluster, either *active (M)*, *passive (B)* or *N/A (#)*.

cluster_peer_dhcp_id

The database identifier (ID) of the other server of the cluster.

cluster_hb_hostaddr

The IP address of the heartbeat of the server, a direct link between the servers of the cluster.

cluster_ssh_keyring_id

The database identifier (ID) of the SSH key dedicated to the cluster communication.

cluster_vip_phys_hostaddr

The local physical IP address of the VIP the cluster relies on.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 14.8. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcp_class_parameters

The class parameters applied to the DHCPv4 server and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcp_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcp_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

dhcp_server_count — Count the number of DHCPv4 servers

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp_server_options_list — List the DHCP options set on a DHCPv4 server

Description

This service allows you to list the DHCP options set on a specific server. To add, edit or delete DHCP options, refer to the chapter [DHCPv4 Option](#).

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcption_id

The database identifier (ID) of the DHCP options set on the DHCPv4 server.

oid

Internal use. Not documented.

row_enabled

The object activation status:

- If set to *0*, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to *1*, the object is enabled and managed.

By default, *row_enabled* is set to *1* when an object is created.

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server.

dhcption_name

The name of the DHCPv4 option.

dhcption_value

The value of the DHCPv4 option.

delayed_create_time

The delay of creation status. *1* indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. *1* indicates that the object is not deleted yet.

modif_count

Internal use. Not documented.

dhcp_name

The name of the DHCPv4 server.

dhcp_type

The type of the DHCPv4 server:

Table 14.9. dhcp_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server. 0 indicates that the server is not managed by a smart architecture or is a smart architecture itself.

Chapter 15. DHCPv6 Server

Name

dhcp6_server6_list — List the DHCPv6 servers

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

ipmdhcp6_https_login

Internal use. Not documented.

connectionprofile_name

The name of the connection profile used as connection method for the DHCPv6 server.

ipmdhcp6_is_package

The DHCPv6 server package information. *Y* for an EfficientIP Package server, *N* for an appliance or virtual machine, *U* the package information is irrelevant. For servers with a *dhcp6_type* set to *ipm*, *U* indicates either EfficientIP Packages or appliances/virtual machines.

isolated

A way to determine if the server can update any other module (1).

tree_path

The database path toward the server as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

vdhcp6_param1

Internal use. Not documented.

snmp_id

Internal use. Not documented.

snmp_port

Internal use. Not documented.

snmp_profile_id

Internal use. Not documented.

snmp_retry

Internal use. Not documented.

snmp_timeout

Internal use. Not documented.

snmp_use_tcp

Internal use. Not documented.

ref1_dhcp6_name

The name of the Master or Single DHCPv6 server within the smart architecture.

vdhcp6_ref1_dhcp6_id

The database identifier (ID) of the DHCPv6 smart architecture the server belongs to.

ref2_dhcp6_name

Internal use. Not documented.

vdhcp6_ref2_dhcp6_id

Internal use. Not documented.

tree_level

The database level of the server. *0* indicates the server is managed on its own, *1* indicates it is managed by a smart architecture.

total_vdhcp6_members

The total number of servers managed by the DHCPv6 smart architecture.

vdhcp6_members_name

The list of the servers managed by the DHCPv6 smart architecture, as follows: <*dhcp6_name*>,<*dhcp6_name*>,... .

vdhcp6_arch

The type of the DHCPv6 smart architecture.

Table 15.1. vdhcp6_arch possible values

| Type | Description |
|------------|---|
| single | The Single-Server smart architecture manages a single DHCPv6 server. |
| splitscope | The Split-Scope smart architecture sets a pair of DHCP servers in a configuration where the two scopes listen to the same subnet, but the range of addresses is divided. |
| stateless | The Stateless smart architecture offers a limited number of options to the DHCP clients. The IP address is delivered thanks to the subnet gateway and it is impossible to create any ranges or statics or to retrieve any leases. |

vdhcp6_parent_name

The name of the DHCPv6 smart architecture managing the DHCPv6 server. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_parent_arch

The type of the DHCPv6 smart architecture managing the DHCPv6 server. No value indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server. *0* indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_ref1_dhcp6_name

Internal use. Not documented.

vdhcp6_ref2_dhcp6_name

Internal use. Not documented.

dhcp6_uboottime

Internal use. Not documented.

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server.

dhcp6_type

The type of the DHCPv6 server:

Table 15.2. dhcp6_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

dhcp6_state

The status of the DHCPv6 server:

Table 15.3. *dhcp6_state* possible values

| Status | Description |
|--------|---|
| ER | The license used in SOLIDserver is not compliant with the added server: the license is invalid. |
| ES | The server configuration could not be parsed properly. |
| ET | The server does not answer anymore due to a scheduled configuration of the server. |
| IS | There was a setting error during the server declaration. For instance, some settings were added to a server that does not support them or a smart architecture is not managing any physical server. |
| IC | The SSL credentials are invalid |
| IP | The account used to add the Microsoft Windows DHCP server does not have sufficient privileges to manage it. |
| LS | The server configuration is not viable. |
| N | The server does not have a status as it has not synchronized yet. |
| Y | The server is operational. |

dhcp6_synching

The synchronization status of the DHCPv6 server. 1 indicates that the server is currently being synchronized.

dhcp6_name

The name of the DHCPv6 server.

dhcp6_comment

The description of the DHCPv6 server.

dhcp6_version

The version details of the DHCPv6 server.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

dhcp6_class_name

The name of the class applied to the DHCPv6 server, it can be preceded by the class directory.

ip_addr

The Management IP address of the DHCPv6 server, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv6 server, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

dhcp6_last_refresh_time

Internal use. Not documented.

stat_enabled

Internal use. Not documented.

stat_period

Internal use. Not documented.

stat_niceness

Internal use. Not documented.

stat_time

Internal use. Not documented.

reverse_proxy_conf

The URL of the HTTP(S) reverse proxy server that forwards client requests to the DHCPv6 server, if you configured one.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 15.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcp6_class_parameters

The class parameters applied to the DHCPv6 server and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

dhcp6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

dhcp6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

Name

dhcp6_server6_info — Display the properties of a DHCPv6 server

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcp6_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

Output Parameters

ipmdhcp6_https_login

Internal use. Not documented.

connectionprofile_name

The name of the connection profile used as connection method for the DHCPv6 server.

ipmdhcp6_is_package

The DHCPv6 server package information. *Y* for an EfficientIP Package server, *N* for an appliance or virtual machine, *U* the package information is irrelevant. For servers with a `dhcp6_type` set to `ipm`, *U* indicates either EfficientIP Packages or appliances/virtual machines.

isolated

A way to determine if the server can update any other module (1).

tree_path

The database path toward the server as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

vdhcp6_param1

Internal use. Not documented.

snmp_id

Internal use. Not documented.

snmp_port

Internal use. Not documented.

snmp_profile_id

Internal use. Not documented.

snmp_retry

Internal use. Not documented.

snmp_timeout

Internal use. Not documented.

snmp_use_tcp

Internal use. Not documented.

ref1_dhcp6_name

The name of the Master or Single DHCPv6 server within the smart architecture.

vdhcp6_ref1_dhcp6_id

The database identifier (ID) of the DHCPv6 smart architecture the server belongs to.

ref2_dhcp6_name

Internal use. Not documented.

vdhcp6_ref2_dhcp6_id

Internal use. Not documented.

tree_level

The database level of the server. 0 indicates the server is managed on its own, 1 indicates it is managed by a smart architecture.

total_vdhcp6_members

The total number of servers managed by the DHCPv6 smart architecture.

vdhcp6_members_name

The list of the servers managed by the DHCPv6 smart architecture, as follows: <dhcp6_name>,<dhcp6_name>,... .

vdhcp6_arch

The type of the DHCPv6 smart architecture.

Table 15.5. vdhcp6_arch possible values

| Type | Description |
|------------|---|
| single | The Single-Server smart architecture manages a single DHCPv6 server. |
| splitscope | The Split-Scope smart architecture sets a pair of DHCP servers in a configuration where the two scopes listen to the same subnet, but the range of addresses is divided. |
| stateless | The Stateless smart architecture offers a limited number of options to the DHCP clients. The IP address is delivered thanks to the subnet gateway and it is impossible to create any ranges or statics or to retrieve any leases. |

vdhcp6_parent_name

The name of the DHCPv6 smart architecture managing the DHCPv6 server. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_parent_arch

The type of the DHCPv6 smart architecture managing the DHCPv6 server. No value indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server. 0 indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_ref1_dhcp6_name

Internal use. Not documented.

vdhcp6_ref2_dhcp6_name

Internal use. Not documented.

dhcp6_uboottime

Internal use. Not documented.

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server.

dhcp6_type

The type of the DHCPv6 server:

Table 15.6. dhcp6_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

dhcp6_state

The status of the DHCPv6 server:

Table 15.7. dhcp6_state possible values

| Status | Description |
|--------|---|
| ER | The license used in SOLIDserver is not compliant with the added server: the license is invalid. |
| ES | The server configuration could not be parsed properly. |
| ET | The server does not answer anymore due to a scheduled configuration of the server. |
| IS | There was a setting error during the server declaration. For instance, some settings were added to a server that does not support them or a smart architecture is not managing any physical server. |
| IC | The SSL credentials are invalid |
| IP | The account used to add the Microsoft Windows DHCP server does not have sufficient privileges to manage it. |
| LS | The server configuration is not viable. |
| N | The server does not have a status as it has not synchronized yet. |
| Y | The server is operational. |

dhcp6_synching

The synchronization status of the DHCPv6 server. 1 indicates that the server is currently being synchronized.

dhcp6_name

The name of the DHCPv6 server.

dhcp6_comment

The description of the DHCPv6 server.

dhcp6_version

The version details of the DHCPv6 server.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

dhcp6_class_name

The name of the class applied to the DHCPv6 server, it can be preceded by the class directory.

ip_addr

The Management IP address of the DHCPv6 server, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv6 server, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

dhcp6_last_refresh_time

Internal use. Not documented.

stat_enabled

Internal use. Not documented.

stat_period

Internal use. Not documented.

stat_niceness

Internal use. Not documented.

stat_time

Internal use. Not documented.

reverse_proxy_conf

The URL of the HTTP(S) reverse proxy server that forwards client requests to the DHCPv6 server, if you configured one.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number> @<multi-status-severity> @<module>. The different severity levels are:

Table 15.8. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcp6_class_parameters

The class parameters applied to the DHCPv6 server and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

dhcp6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

dhcp6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

Name

dhcp6_server6_count — Count the number of DHCPv6 servers

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp6_server6_options6_list — List the DHCP options set on a DHCPv6 server

Description

This service allows you to list the DHCP options set on a specific server. To add, edit or delete DHCP options, refer to the chapter [DHCPv6 Option](#).

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcption6_id

The database identifier (ID) of the DHCP options set on the DHCPv6 server.

oid

Internal use. Not documented.

row_enabled

The object activation status:

- If set to *0*, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to *1*, the object is enabled and managed.

By default, *row_enabled* is set to *1* when an object is created.

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server.

dhcption6_name

The name of the DHCPv6 option.

dhcption6_value

The value of the DHCPv6 option.

delayed_time

The delay of creation/deletion status. *1* indicates that the object is not created/deleted yet.

modif_count

Internal use. Not documented.

dhcp6_name

The name of the DHCPv6 server.

dhcp6_type

The type of the DHCPv6 server:

Table 15.9. *dhcp6_type* possible values

| Type | Description |
|------|---|
| ipm | EfficientIP or EfficientIP Package server |

| Type | Description |
|-------|---------------------------------------|
| vdhcp | EfficientIP DHCPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server. *0* indicates that the server is not managed by a smart architecture or is a smart architecture itself.

Chapter 16. DHCPv4 Shared Network

Name

dhcp_sn_add — Add/Edit a DHCPv4 shared network

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcpsn_name && (dhcp_id || dhcp_name || hostaddr))
- **Editing:** ((dhcpsn_id || dhcpsn_name) && (dhcp_id || dhcp_name || hostaddr))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network, a unique numeric key value automatically incremented when you add a DHCPv4 shared network. Use the ID to specify the DHCPv4 shared network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpsn_name

The name of the DHCPv4 shared network, each DHCPv4 shared network must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp_shared_network_list — List the DHCPv4 shared networks

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network.

dhcpsn_name

The name of the DHCPv4 shared network.

delayed_create_time

The delay of creation status. *1* indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. *1* indicates that the object is not deleted yet.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 16.1. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. *0* indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

row_enabled

The object activation status:

- If set to *0*, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to *1*, the object is enabled and managed.

By default, *row_enabled* is set to *1* when an object is created.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 16.2. Multi-status severity levels

| Message number | Severity | Description |
|-----------------------|-----------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

Name

dhcp_shared_network_info — Display the properties of a DHCPv4 shared network

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpsn_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network, a unique numeric key value automatically incremented when you add a DHCPv4 shared network. Use the ID to specify the DHCPv4 shared network of your choice.

Output Parameters

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network.

dhcpsn_name

The name of the DHCPv4 shared network.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 16.3. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. *0* indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

row_enabled

The object activation status:

- If set to *0*, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to *1*, the object is enabled and managed.

By default, *row_enabled* is set to *1* when an object is created.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 16.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

Name

dhcp_shared_network_count — Count the number of DHCPv4 shared networks

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp_sn_delete — Delete a DHCPv4 Shared Network

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((dhcpsn_id || dhcpsn_name) && (dhcp_id || dhcp_name || hostaddr))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network, a unique numeric key value automatically incremented when you add a DHCPv4 shared network. Use the ID to specify the DHCPv4 shared network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpsn_name

The name of the DHCPv4 shared network.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 17. DHCPv6 Shared Network

Name

dhcp6_sn6_add — Add/Edit a DHCPv6 shared network

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcpsn6_name && (dhcp6_id || dhcp6_name || hostaddr))
- **Editing:** ((dhcpsn6_id || dhcpsn6_name) && (dhcp6_id || dhcp6_name || hostaddr))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpsn6_id

The database identifier (ID) of the DHCPv6 shared network, a unique numeric key value automatically incremented when you add a DHCPv6 shared network. Use the ID to specify the DHCPv6 shared network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpsn6_name

The name of the DHCPv6 shared network.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp6_shared_network6_list — List the DHCPv6 shared networks

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter **offset** must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter **limit** must be specified in **lowercase**.

Output Parameters

dhcpsn6_id

The database identifier (ID) of the DHCPv6 shared network.

dhcpsn6_name

The name of the DHCPv6 shared network.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 17.1. *dhcp6_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_arch

The type of the DHCPv6 smart architecture the object belongs to.

Table 17.2. *vdhcp6_arch* possible values

| Type | Description |
|------------|---|
| single | The Single-Server smart architecture manages a single DHCPv6 server. |
| splitscope | The Split-Scope smart architecture sets a pair of DHCP servers in a configuration where the two scopes listen to the same subnet, but the range of addresses is divided. |
| stateless | The Stateless smart architecture offers a limited number of options to the DHCP clients. The IP address is delivered thanks to the subnet gateway and it is impossible to create any ranges or statics or to retrieve any leases. |

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 17.3. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

Name

dhcp6_shared_network6_info — Display the properties of a DHCPv6 shared network

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpsn6_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpsn6_id

The database identifier (ID) of the DHCPv6 shared network, a unique numeric key value automatically incremented when you add a DHCPv6 shared network. Use the ID to specify the DHCPv6 shared network of your choice.

Output Parameters

dhcpsn6_id

The database identifier (ID) of the DHCPv6 shared network.

dhcpsn6_name

The name of the DHCPv6 shared network.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 17.4. dhcp6_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_arch

The type of the DHCPv6 smart architecture the object belongs to.

Table 17.5. vdhcp6_arch possible values

| Type | Description |
|------------|---|
| single | The Single-Server smart architecture manages a single DHCPv6 server. |
| splitscope | The Split-Scope smart architecture sets a pair of DHCP servers in a configuration where the two scopes listen to the same subnet, but the range of addresses is divided. |
| stateless | The Stateless smart architecture offers a limited number of options to the DHCP clients. The IP address is delivered thanks to the subnet gateway and it is impossible to create any ranges or statics or to retrieve any leases. |

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 17.6. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

Name

dhcp6_shared_network6_count—Count the number of DHCPv6 shared networks

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp6_sn6_delete — Delete a DHCPv6 Shared Network

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((dhcpsn6_id || dhcpsn6_name) && (dhcp6_id || dhcp6_name || hostaddr))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_addr

Deprecated, replaced by **hostaddr**.

dhcpsn6_id

The database identifier (ID) of the DHCPv6 shared network, a unique numeric key value automatically incremented when you add a DHCPv6 shared network. Use the ID to specify the DHCPv6 shared network of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpsn6_name

The name of the DHCPv6 shared network.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 18. DHCPv4 Scope

Name

dhcp_scope_add — Add/Edit a DHCPv4 scope

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcpscope_net_addr && dhcpscope_net_mask && (dhcp_id || dhcp_name || hostaddr))
- **Editing:** (dhcpscope_id || (dhcpscope_net_addr && dhcpscope_net_mask && (dhcp_id || dhcp_name || hostaddr)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

netaddr

Deprecated, replaced by **dhcpscope_net_addr**.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope.

| Type | IPv4 address | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | No |

dhcpscope_start_addr

Deprecated, replaced by **dhcpscope_net_addr**.

dhcpscope_netaddr

Deprecated, replaced by **dhcpscope_net_addr**.

netmask

Deprecated, replaced by **dhcpscope_net_mask**.

dhcpscope_net_mask

The netmask of the DHCPv4 scope. It is expressed in dot-decimal notation and defines the number of addresses the scope contains.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

dhcpscope_netmask

Deprecated, replaced by **dhcpscope_net_mask**.

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network, a unique numeric key value automatically incremented when you add a DHCPv4 shared network. Use the ID to specify the DHCPv4 shared network of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dhcpsn_name

The name of the DHCPv4 shared network.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope, a unique numeric key value automatically incremented when you add a DHCPv4 scope. Use the ID to specify the DHCPv4 scope of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope_name

The name of the DHCPv4 scope, each DHCPv4 scope must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dhcpfailover_id

The database identifier (ID) of the DHCPv4 failover channel, a unique numeric key value automatically incremented when you add a DHCPv4 failover channel. Use the ID to specify the DHCPv4 failover channel of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dhcpfailover_name

The name of the DHCPv4 failover channel.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dhcpscope_site_id

The database identifier (ID) of an existing space you want to associate with the DHCPv4 scope.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dhcpscope_site_name

The name of an existing space you want to associate with the DHCPv4 scope.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dhcpscope_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

dhcpscope_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: **<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....**. If the inheritance or propagation property is not specified, its default value - *set, propagate* - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp_scope_list — List the DHCPv4 scopes

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

vdhcp_arch

The type of the DHCPv4 smart architecture the object belongs to.

Table 18.1. *vdhcp_arch* possible values

| Type | Description |
|-------------|--|
| masterslave | The One-to-One smart architecture sets a pair of DHCP servers in a Master/Backup configuration. |
| star | The One-to-Many smart architecture sets a multi-site failover configuration at the cost of n-servers+1. |
| splitscope | The Split-Scope smart architecture sets a pair of DHCP servers in a configuration where the two scopes listen to the same subnet, but the range of addresses is divided. |
| single | The Single-Server smart architecture manages a single DHCP server. |

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 18.2. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

dhcpfailover_id

The database identifier (ID) of the DHCPv4 failover channel associated with the object.

dhcpfailover_name

The name of the DHCPv4 failover channel associated with the object.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcpscope_name

The name of the DHCPv4 scope.

dhcpscope_start_ip_addr

The first IP address of the DHCPv4 scope, in hexadecimal format.

dhcpscope_end_ip_addr

The last IP address of the DHCPv4 scope, in hexadecimal format.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope.

dhcpscope_net_mask

The netmask of the DHCPv4 scope. It is expressed in dot-decimal notation and defines the number of addresses the scope contains.

dhcpscope_size

The number of IP addresses the DHCPv4 scope contains.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

dhcpscope_site_name

The name of the space associated with the DHCPv4 scope.

dhcpscope_site_id

The database identifier (ID) of the space associated with the DHCPv4 scope.

dhcpscope_sort_name

Internal use. Not documented.

dhcpscope_class_name

The name of the class applied to the DHCPv4 scope, it can be preceded by the class directory.

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network the object belongs to.

dhcpsn_name

The name of the DHCPv4 shared network the object belongs to.

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. #indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp_class_name

The name of the class applied to the DHCPv4 server the object belongs to, it can be preceded by the class directory.

dhcp_version

The version details of the DHCPv4 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

ip_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 18.3. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcpscope_class_parameters

The class parameters applied to the DHCPv4 scope and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpscope_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcpscope_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcp_class_parameters

The class parameters applied to the DHCPv4 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Example

In the example below, we call the service **dhcp_scope_list** with PHP (cURL) using the clause **WHERE** to return the scopes which class parameter *information* is *important* or the scopes which class parameter *description* contains *accounting*. For more details regarding the use of class parameters in the clause, refer to the chapter [Calling Services With TAGS](#).

Example 18.1. Calling the service dhcp_scope_list using PHP and WHERE

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/dhcp_scope_list?TAGS=" .
    "dhcpscope.information%26dhcpscope.description&WHERE=" .
    "+tag_dhcpscope_information%20like%20%27important%27%20or%20tag_dhcpscope_description%20like%20%27%25accounting%25%27",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Name

dhcp_scope_info — Display the properties of a DHCPv4 scope

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpscope_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope, a unique numeric key value automatically incremented when you add a DHCPv4 scope. Use the ID to specify the DHCPv4 scope of your choice.

Output Parameters

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

vdhcp_arch

The type of the DHCPv4 smart architecture the object belongs to.

Table 18.4. vdhcp_arch possible values

| Type | Description |
|-------------|---|
| masterslave | The One-to-One smart architecture sets a pair of DHCP servers in a Master/Backup configuration. |

| Type | Description |
|------------|--|
| star | The One-to-Many smart architecture sets a multi-site failover configuration at the cost of n-servers+1. |
| splitscope | The Split-Scope smart architecture sets a pair of DHCP servers in a configuration where the two scopes listen to the same subnet, but the range of addresses is divided. |
| single | The Single-Server smart architecture manages a single DHCP server. |

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 18.5. dhcp_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

dhcpfailover_id

The database identifier (ID) of the DHCPv4 failover channel associated with the object.

dhcpfailover_name

The name of the DHCPv4 failover channel associated with the object.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcpscope_name

The name of the DHCPv4 scope.

dhcpscope_start_ip_addr

The first IP address of the DHCPv4 scope, in hexadecimal format.

dhcpscope_end_ip_addr

The last IP address of the DHCPv4 scope, in hexadecimal format.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope.

dhcpscope_net_mask

The netmask of the DHCPv4 scope. It is expressed in dot-decimal notation and defines the number of addresses the scope contains.

dhcpscope_size

The number of IP addresses the DHCPv4 scope contains.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

dhcpscope_site_name

The name of the space associated with the DHCPv4 scope.

dhcpscope_site_id

The database identifier (ID) of the space associated with the DHCPv4 scope.

dhcpscope_sort_name

Internal use. Not documented.

dhcpscope_class_name

The name of the class applied to the DHCPv4 scope, it can be preceded by the class directory.

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network the object belongs to.

dhcpsn_name

The name of the DHCPv4 shared network the object belongs to.

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp_class_name

The name of the class applied to the DHCPv4 server the object belongs to, it can be preceded by the class directory.

dhcp_version

The version details of the DHCPv4 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

ip_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 18.6. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcpscope_class_parameters

The class parameters applied to the DHCPv4 scope and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpscope_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcpscope_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcp_class_parameters

The class parameters applied to the DHCPv4 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dhcp_scope_count — Count the number of DHCPv4 scopes

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp_scope_groupby — Group DHCPv4 scopes by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

dhcp_scope_groupby_count — Count the number of DHCPv4 scopes grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

dhcp_scope_options_list — List the DHCP options set on a DHCPv4 scope

Description

This service allows you to list the DHCP options set on a specific scope. To add, edit or delete DHCP options, refer to the chapter [DHCPv4 Option](#).

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope, a unique numeric key value automatically incremented when you add a DHCPv4 scope. Use the ID to specify the DHCPv4 scope of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcpscopecoption_id

The database identifier (ID) of the DHCP option set on the DHCPv4 scope.

oid

Internal use. Not documented.

row_enabled

The object activation status:

- If set to *0*, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to *1*, the object is enabled and managed.

By default, *row_enabled* is set to *1* when an object is created.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope.

dhcption_name

The name of the DHCPv4 option.

dhcption_value

The value of the DHCPv4 option.

delayed_create_time

The delay of creation status. *1* indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. *1* indicates that the object is not deleted yet.

modif_count

Internal use. Not documented.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 18.7. dhcp_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpscope_name

The name of the DHCPv4 scope.

dhcpscope_if_name

Internal use. Not documented.

dhcpscope_if_addr

Internal use. Not documented.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope.

dhcpscope_net_mask

The netmask of the DHCPv4 scope. It is expressed in dot-decimal notation and defines the number of addresses the scope contains.

Name

group_dhcpscope_add — Add a DHCPv4 scope to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (dhcpscope_id || (dhcpscope_net_addr && (dhcp_id || dhcp_name || hostaddr))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope, a unique numeric key value automatically incremented when you add a DHCPv4 scope. Use the ID to specify the DHCPv4 scope of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

scope_id

Deprecated, replaced by **dhcpscope_id**.

netaddr

Deprecated, replaced by **dhcpscope_net_addr**.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope_start_addr

Deprecated, replaced by **dhcpscope_net_addr**.

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

group_dhcpscope_delete — Remove a DHCPv4 scope from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (dhcpscope_id || (dhcpscope_net_addr && (dhcp_id || dhcp_name || hostaddr))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope, a unique numeric key value automatically incremented when you add a DHCPv4 scope. Use the ID to specify the DHCPv4 scope of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

scope_id

Deprecated, replaced by **dhcpscope_id**.

netaddr

Deprecated, replaced by **dhcpscope_net_addr**.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope_start_addr

Deprecated, replaced by **dhcpscope_net_addr**.

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

dhcp_scope_delete — Delete a DHCPv4 scope

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcpscope_id || (dhcpscope_net_addr && (dhcp_id || dhcp_name || hostaddr)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope, a unique numeric key value automatically incremented when you add a DHCPv4 scope. Use the ID to specify the DHCPv4 scope of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

scope_id

Deprecated, replaced by **dhcpscope_id**.

netaddr

Deprecated, replaced by **dhcpscope_net_addr**.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope_start_addr

Deprecated, replaced by **dhcpscope_net_addr**.

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 19. DHCPv6 Scope

Name

dhcp6_scope6_add — Add/Edit a DHCPv6 scope

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcpscope6_start_addr && (dhcpscope6_end_addr || dhcpscope6_prefix) && (dhcp6_id || dhcp6_name || hostaddr))
- **Editing:** (dhcpscope6_id || (dhcpscope6_start_addr && (dhcpscope6_end_addr || dhcpscope6_prefix) && (dhcp6_id || dhcp6_name || hostaddr)))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpscope6_start_addr

The first IP address of the DHCPv6 scope.

| Type | IPv6 address | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | No |

dhcpscope6_end_addr

The last IP address of the DHCPv6 scope.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

dhcpscope6_prefix

The prefix of the DHCPv6 scope, an integer that defines the number of address the scope contains.

| | | | |
|---------------|---|----------------|-----|
| Type | IPv6 prefix (integer between 1 and 128) | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope, a unique numeric key value automatically incremented when you add a DHCPv6 scope. Use the ID to specify the DHCPv6 scope of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope6_name

The name of the DHCPv6 scope, each DHCPv6 scope must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dhcpfailover6_id

The database identifier (ID) of the DHCPv6 failover channel, a unique numeric key value automatically incremented when you add a DHCPv6 failover channel. Use the ID to specify the DHCPv6 failover channel of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dhcpfailover6_name

The name of the DHCPv6 failover channel.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dhcpscope6_site_id

The database identifier (ID) of an existing space you want to associate with the DHCPv6 scope.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dhcpscope6_site_name

The name of an existing space you want to associate with the DHCPv6 scope.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dhcpscope6_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

dhcpscope6_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope6_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>_class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... . If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp6_scope6_list — List the DHCPv6 scopes

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_arch

The type of the DHCPv6 smart architecture the object belongs to.

Table 19.1. *vdhcp6_arch* possible values

| Type | Description |
|------------|---|
| single | The Single-Server smart architecture manages a single DHCPv6 server. |
| splitscope | The Split-Scope smart architecture sets a pair of DHCP servers in a configuration where the two scopes listen to the same subnet, but the range of addresses is divided. |
| stateless | The Stateless smart architecture offers a limited number of options to the DHCP clients. The IP address is delivered thanks to the subnet gateway and it is impossible to create any ranges or statics or to retrieve any leases. |

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 19.2. *dhcp6_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

dhcpfailover6_id

The database identifier (ID) of the DHCPv6 failover channel associated with the object.

dhcpfailover6_name

The name of the DHCPv6 failover channel associated with the object.

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcpscope6_name

The name of the DHCIPv6 scope.

dhcpscope6_start_ip6_addr

The first IP address of the DHCIPv6 scope, in hexadecimal format.

dhcpscope6_end_ip6_addr

The last IP address of the DHCIPv6 scope, in hexadecimal format.

dhcpscope6_size

The number of IP addresses the DHCIPv6 scope contains.

dhcpscope6_prefix

The prefix of the DHCIPv6 scope.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

dhcpscope6_site_name

The name of the space associated with the DHCIPv6 scope.

dhcpscope6_site_id

The database identifier (ID) of the space associated with the DHCIPv6 scope.

dhcpscope6_sort_name

Internal use. Not documented.

dhcpscope6_class_name

The name of the class applied to the DHCIPv6 scope, it can be preceded by the class directory.

dhcpsn6_id

The database identifier (ID) of the DHCIPv6 shared network the object belongs to.

dhcpsn6_name

The name of the DHCIPv6 shared network the object belongs to.

vdhcp6_parent_name

The name of the DHCIPv4 smart architecture managing the DHCIPv4 server the object belongs to. #indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp6_class_name

The name of the class applied to the DHCIPv6 server the object belongs to, it can be preceded by the class directory.

dhcp6_version

The version details of the DHCIPv6 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

ip_addr

The Management IP address of the DHCIPv6 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCIPv6 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 19.3. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcpscope6_class_parameters

The class parameters applied to the DHCPv6 scope and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpscope6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcpscope6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcp6_class_parameters

The class parameters applied to the DHCPv6 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dhcp6_scope6_info — Display the properties of a DHCPv6 scope

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpscope6_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope, a unique numeric key value automatically incremented when you add a DHCPv6 scope. Use the ID to specify the DHCPv6 scope of your choice.

Output Parameters

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_arch

The type of the DHCPv6 smart architecture the object belongs to.

Table 19.4. `vdhcp6_arch` possible values

| Type | Description |
|--------|--|
| single | The Single-Server smart architecture manages a single DHCPv6 server. |

| Type | Description |
|------------|---|
| splitscope | The Split-Scope smart architecture sets a pair of DHCP servers in a configuration where the two scopes listen to the same subnet, but the range of addresses is divided. |
| stateless | The Stateless smart architecture offers a limited number of options to the DHCP clients. The IP address is delivered thanks to the subnet gateway and it is impossible to create any ranges or statics or to retrieve any leases. |

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 19.5. dhcp6_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

dhcpfailover6_id

The database identifier (ID) of the DHCPv6 failover channel associated with the object.

dhcpfailover6_name

The name of the DHCPv6 failover channel associated with the object.

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcpscope6_name

The name of the DHCPv6 scope.

dhcpscope6_start_ip6_addr

The first IP address of the DHCPv6 scope, in hexadecimal format.

dhcpscope6_end_ip6_addr

The last IP address of the DHCPv6 scope, in hexadecimal format.

dhcpscope6_size

The number of IP addresses the DHCPv6 scope contains.

dhcpscope6_prefix

The prefix of the DHCPv6 scope.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

dhcpscope6_site_name

The name of the space associated with the DHCPv6 scope.

dhcpscope6_site_id

The database identifier (ID) of the space associated with the DHCPv6 scope.

dhcpscope6_sort_name

Internal use. Not documented.

dhcpscope6_class_name

The name of the class applied to the DHCPv6 scope, it can be preceded by the class directory.

dhcpsn6_id

The database identifier (ID) of the DHCPv6 shared network the object belongs to.

dhcpsn6_name

The name of the DHCPv6 shared network the object belongs to.

vdhcp6_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp6_class_name

The name of the class applied to the DHCPv6 server the object belongs to, it can be preceded by the class directory.

dhcp6_version

The version details of the DHCPv6 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

ip_addr

The Management IP address of the DHCPv6 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv6 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 19.6. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcpscope6_class_parameters

The class parameters applied to the DHCPv6 scope and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpscope6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcpscope6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcp6_class_parameters

The class parameters applied to the DHCPv6 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dhcp6_scope6_count — Count the number of DHCPv6 scopes

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp6_scope6_options6_list — List the DHCP options set on a DHCPv6 scope

Description

This service allows you to list the DHCP options set on a specific scope. To add, edit or delete DHCP options, refer to the chapter [DHCPv6 Option](#).

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope, a unique numeric key value automatically incremented when you add a DHCPv6 scope. Use the ID to specify the DHCPv6 scope of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>' or <parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcpscopecoption6_id

The database identifier (ID) of the DHCP option set on the DHCPv6 scope.

oid

Internal use. Not documented.

row_enabled

The object activation status:

- If set to *0*, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to *1*, the object is enabled and managed.

By default, *row_enabled* is set to *1* when an object is created.

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope.

dhcoption6_name

The name of the DHCPv6 option.

dhcoption6_value

The value of the DHCPv6 option.

delayed_time

The delay of creation/deletion status. *1* indicates that the object is not created/deleted yet.

modif_count

Internal use. Not documented.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 19.7. dhcp6_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpscope6_name

The name of the DHCPv6 scope.

dhcpscope6_start_ip6_addr

The first IP address of the DHCPv6 scope, in hexadecimal format.

dhcpscope6_end_ip6_addr

The last IP address of the DHCPv6 scope, in hexadecimal format.

Name

group_dhcpscope6_add — Add a DHCPv6 scope to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (dhcpscope6_id || (dhcpscope6_start_addr && (dhcp6_id || dhcp6_name || hostaddr))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope, a unique numeric key value automatically incremented when you add a DHCPv6 scope. Use the ID to specify the DHCPv6 scope of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope6_start_addr

The first IP address of the DHCPv6 scope.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

group_dhcpscope6_delete — Remove a DHCPv6 scope from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (dhcpscope6_id || (dhcpscope6_start_addr && (dhcp6_id || dhcp6_name || hostaddr))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope, a unique numeric key value automatically incremented when you add a DHCPv6 scope. Use the ID to specify the DHCPv6 scope of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope6_start_addr

The first IP address of the DHCPv6 scope.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

dhcp6_scope6_delete — Delete a DHCPv6 scope

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcpscope6_id || (dhcpscope6_start_addr && (dhcp6_id || dhcp6_name || hostaddr)))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope, a unique numeric key value automatically incremented when you add a DHCPv6 scope. Use the ID to specify the DHCPv6 scope of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpscope6_start_addr

The first IP address of the DHCPv6 scope.

| Type | IPv6 address | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 20. DHCPv4 Group

Name

dhcp_group_add — Add a DHCPv4 group

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcpgroup_name && (dhcp_id || dhcp_name || hostaddr))
- **Editing:** (dhcpgroup_id || (dhcpgroup_name && (dhcp_id || dhcp_name || hostaddr)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpgroup_name

The name of the DHCPv4 group, each DHCPv4 group must have a unique name.

| Type | String | Maximum length | 64 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpgroup_id

The database identifier (ID) of the DHCPv4 group, a unique numeric key value automatically incremented when you add a DHCPv4 group. Use the ID to specify the DHCPv4 group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpgroup_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

dhcpgroup_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: *<class-parameter1>=<value1>&<class-parameter2>=<value2>&... .*

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: *<class-parameter1>&<class-parameter2>&... .* Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpgroup_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .* If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp_group_list — List the DHCPv4 groups

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcpgroup_id

The database identifier (ID) of the DHCPv4 group.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 20.1. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpgroup_name

The name of the DHCPv4 group.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp_class_name

The name of the class applied to the DHCPv4 server the object belongs to, it can be preceded by the class directory.

dhcp_version

The version details of the DHCPv4 server the object belongs to.

ip_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

dhcpgroup_class_name

The name of the class applied to the DHCPv4 group, it can be preceded by the class directory.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number> @<multi-status-severity> @<module>. The different severity levels are:

Table 20.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcpgroup_class_parameters

The class parameters applied to the DHCPv4 group and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpgroup_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpgroup_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcpgroup_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcp_class_parameters

The class parameters applied to the DHCPv4 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dhcp_group_info — Display the properties of a DHCPv4 group

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpgroup_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

dhcpgroup_id

The database identifier (ID) of the DHCPv4 group, a unique numeric key value automatically incremented when you add a DHCPv4 group. Use the ID to specify the DHCPv4 group of your choice.

Output Parameters

dhcpgroup_id

The database identifier (ID) of the DHCPv4 group.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 20.3. dhcp_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpgroup_name

The name of the DHCPv4 group.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp_class_name

The name of the class applied to the DHCPv4 server the object belongs to, it can be preceded by the class directory.

dhcp_version

The version details of the DHCPv4 server the object belongs to.

ip_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

dhcpgroup_class_name

The name of the class applied to the DHCPv4 group, it can be preceded by the class directory.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 20.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|---|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcpgroup_class_parameters

The class parameters applied to the DHCPv4 group and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpgroup_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpgroup_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcpgroup_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcp_class_parameters

The class parameters applied to the DHCPv4 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dhcp_group_count — Count the number of DHCPv4 groups

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp_group_delete — Delete a DHCPv4 group

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcpgroup_id || (dhcpgroup_name && (dhcp_id || dhcp_name || hostaddr)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpgroup_id

The database identifier (ID) of the DHCPv4 group, a unique numeric key value automatically incremented when you add a DHCPv4 group. Use the ID to specify the DHCPv4 group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

group_id

Deprecated, replaced by **dhcpgroup_id**.

dhcpgroup_name

The name of the DHCPv4 group.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

group_name

Deprecated, replaced by **dhcpgroup_name**.

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 21. DHCPv6 Group

Name

dhcp6_group6_list — List the DHCPv6 groups

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcpgroup6_id

The database identifier (ID) of the DHCPv6 group.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 21.1. *dhcp6_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpgroup6_name

The name of the DHCPv6 group.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

vdhcp6_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp6_class_name

The name of the class applied to the DHCPv6 server the object belongs to, it can be preceded by the class directory.

dhcp6_version

The version details of the DHCPv6 server the object belongs to.

ip_addr

The Management IP address of the DHCIPv6 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCIPv6 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

dhcpgroup6_class_name

The name of the class applied to the DHCIPv6 group, it can be preceded by the class directory.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 21.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcpgroup6_class_parameters

The class parameters applied to the DHCIPv6 group and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpgroup6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpgroup6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcpgroup6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcp6_class_parameters

The class parameters applied to the DHCIPv6 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Chapter 22. DHCPv4 Range

Name

dhcp_range_add — Add/Edit a DHCPv4 range

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcprange_start_addr && dhcprange_end_addr && (dhcpscope_id || dhcp_id || dhcp_name || hostaddr))
- **Editing:** (dhcprange_id || (dhcprange_start_addr && dhcprange_end_addr && (dhcpscope_id || dhcp_id || dhcp_name || hostaddr)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpscope_name

The name of the DHCPv4 scope.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope, a unique numeric key value automatically incremented when you add a DHCPv4 scope. Use the ID to specify the DHCPv4 scope of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

scope_id

Deprecated, replaced by **dhcpscope_id**.

dhcprange_id

The database identifier (ID) of the DHCPv4 range, a unique numeric key value automatically incremented when you add a DHCPv4 range. Use the ID to specify the DHCPv4 range of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

start_addr

Deprecated, replaced by **dhcprange_start_addr**.

dhcprange_start_addr

The first IP address of the DHCPv4 range.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

Deprecated, replaced by **dhcprange_end_addr**.

dhcprange_end_addr

The last IP address of the DHCPv4 range.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

acl

Deprecated, replaced by **dhcprange_acl**.

dhcprange_name

The start and end IP address of the DHCPv4 range, as follows: <start-ip>-<end-ip>.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 32 |
| Default value | N/A | Can be edited | Yes |

dhcprange_acl

The list of ACLs associated with the DHCPv4 range, as follows: <ACL_name>;<ACL_name>;...

| | | | |
|---------------|--------|----------------|------|
| Type | String | Maximum length | 4000 |
| Default value | N/A | Can be edited | Yes |

dhcprange_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

dhcprange_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcprange_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>_class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... . If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp_range_list — List the DHCPv4 ranges

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcprange_id

The database identifier (ID) of the DHCPv4 range.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope the object belongs to.

dhcprange_name

The start and end IP address of the DHCPv4 range, **dhcprange_start_addr** and **dhcprange_end_addr**, as follows: <start-ip>-<end-ip>.

dhcprange_start_addr

The first IP address of the DHCPv4 range.

dhcprange_end_addr

The last IP address of the DHCPv4 range.

dhcprange_start_ip_addr

The first IP address of the DHCPv4 range, in hexadecimal format.

dhcprange_end_ip_addr

The last IP address of the DHCPv4 range, in hexadecimal format.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcpscope_site_id

The database identifier (ID) of the space associated with the DHCPv4 scope the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 22.1. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

dhcp_name

The name of the DHCPv4 server the object belongs to.

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpscope_name

The name of the DHCPv4 scope the object belongs to.

dhcpscope_if_name

Internal use. Not documented.

dhcpscope_if_addr

Internal use. Not documented.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope the object belongs.

dhcpscope_net_mask

The netmask of the DHCPv4 scope the object belongs to. It is expressed in dot-decimal notation and defines the number of addresses the scope contains.

dhcpscope_start_ip_addr

The first IP address of the DHCPv4 scope the object belongs to, in hexadecimal format.

dhcpscope_size

The number of IP addresses the DHCPv4 scope the object belongs to contains.

dhcprange_failover_name

Internal use. Not documented.

dhcprange_state

Internal use. Not documented.

dhcprange_class_name

The name of the class applied to the DHCPv4 range, it can be preceded by the class directory.

dhcprange_lease_count

The total number of leases currently delivered by the DHCPv4 range.

dhcprange_size

The number of IP addresses the DHCPv4 range contains.

dhcprange_lease_percent

The percentage of leases currently delivered by the DHCPv4 range.

dhcprange_acl

The list of ACLs associated with the DHCPv4 range, as follows: <ACL_name>;<ACL_name>;...

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network the object belongs to.

dhcpsn_name

The name of the DHCPv4 shared network the object belongs to.

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. #indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp_comment

The description of the DHCPv4 server the object belongs to.

dhcpscope_class_name

The name of the class applied to the DHCPv4 scope the object belongs to, it can be preceded by the class directory.

dhcp_class_name

The name of the class applied to the DHCPv4 server the object belongs to, it can be preceded by the class directory.

dhcp_version

The version details of the DHCPv4 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

ip_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 22.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcprange_class_parameters

The class parameters applied to the DHCPv4 range and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcprange_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcprange_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcprange_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcpscope_class_parameters

The class parameters applied to the DHCPv4 scope the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcpscope_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcp_class_parameters

The class parameters applied to the DHCPv4 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcp_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dhcp_range_info — Display the properties of a DHCPv4 range

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcprange_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

dhcprange_id

The database identifier (ID) of the DHCPv4 range, a unique numeric key value automatically incremented when you add a DHCPv4 range. Use the ID to specify the DHCPv4 range of your choice.

Output Parameters

dhcprange_id

The database identifier (ID) of the DHCPv4 range.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope the object belongs to.

dhcprange_name

The start and end IP address of the DHCPv4 range, **dhcprange_start_addr** and **dhcprange_end_addr**, as follows: <start-ip>-<end-ip>.

dhcprange_start_addr

The first IP address of the DHCPv4 range.

dhcprange_end_addr

The last IP address of the DHCPv4 range.

dhcprange_start_ip_addr

The first IP address of the DHCPv4 range, in hexadecimal format.

dhcprange_end_ip_addr

The last IP address of the DHCPv4 range, in hexadecimal format.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcpscope_site_id

The database identifier (ID) of the space associated with the DHCPv4 scope the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 22.3. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

dhcp_name

The name of the DHCPv4 server the object belongs to.

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpscope_name

The name of the DHCPv4 scope the object belongs to.

dhcpscope_if_name

Internal use. Not documented.

dhcpscope_if_addr

Internal use. Not documented.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope the object belongs.

dhcpscope_net_mask

The netmask of the DHCPv4 scope the object belongs to. It is expressed in dot-decimal notation and defines the number of addresses the scope contains.

dhcpscope_start_ip_addr

The first IP address of the DHCPv4 scope the object belongs to, in hexadecimal format.

dhcpscope_size

The number of IP addresses the DHCPv4 scope the object belongs to contains.

dhcprange_failover_name

Internal use. Not documented.

dhcprange_state

Internal use. Not documented.

dhcprange_class_name

The name of the class applied to the DHCPv4 range, it can be preceded by the class directory.

dhcprange_lease_count

The total number of leases currently delivered by the DHCPv4 range.

dhcprange_size

The number of IP addresses the DHCPv4 range contains.

dhcprange_lease_percent

The percentage of leases currently delivered by the DHCPv4 range.

dhcprange_acl

The list of ACLs associated with the DHCPv4 range, as follows: <ACL_name>;<ACL_name>;...

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network the object belongs to.

dhcpsn_name

The name of the DHCPv4 shared network the object belongs to.

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. #indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp_comment

The description of the DHCPv4 server the object belongs to.

dhcpscope_class_name

The name of the class applied to the DHCPv4 scope the object belongs to, it can be preceded by the class directory.

dhcp_class_name

The name of the class applied to the DHCPv4 server the object belongs to, it can be preceded by the class directory.

dhcp_version

The version details of the DHCPv4 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

ip_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 22.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcprange_class_parameters

The class parameters applied to the DHCPv4 range and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcprange_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcprange_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcprange_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcpscope_class_parameters

The class parameters applied to the DHCPv4 scope the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpscope_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcp_class_parameters

The class parameters applied to the DHCPv4 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dhcp_range_count — Count the number of DHCPv4 ranges

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp_range_options_list — List the DHCP options set on a DHCPv4 range

Description

This service allows you to list the DHCP options set on a specific range. To add, edit or delete DHCP options, refer to the chapter [DHCPv4 Option](#).

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcprange_id

The database identifier (ID) of the DHCPv4 range, a unique numeric key value automatically incremented when you add a DHCPv4 range. Use the ID to specify the DHCPv4 range of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>' or <parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcprangeoption_id

The database identifier (ID) of the DHCP option set on the DHCPv4 range.

oid

Internal use. Not documented.

row_enabled

The object activation status:

- If set to *0*, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to *1*, the object is enabled and managed.

By default, *row_enabled* is set to *1* when an object is created.

dhcprange_id

The database identifier (ID) of the DHCPv4 range.

dhcpoption_name

The name of the DHCPv4 option.

dhcpoption_value

The value of the DHCPv4 option.

delayed_create_time

The delay of creation status. *1* indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. *1* indicates that the object is not deleted yet.

modif_count

Internal use. Not documented.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 22.5. dhcp_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

Name

dhcp_range_delete — Delete a DHCPv4 range

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcprange_id || ((dhcprange_name || dhcprange_start_addr || dhcprange_end_addr) && (dhcp_id || dhcp_name || hostaddr || dhcpscope_id)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope, a unique numeric key value automatically incremented when you add a DHCPv4 scope. Use the ID to specify the DHCPv4 scope of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

scope_id

Deprecated, replaced by **dhcpscope_id**.

dhcprange_id

The database identifier (ID) of the DHCPv4 range, a unique numeric key value automatically incremented when you add a DHCPv4 range. Use the ID to specify the DHCPv4 range of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

range_id

Deprecated, replaced by **dhcprange_id**.

dhcprange_name

The start and end IP address of the DHCPv4 range, as follows: <start-ip>-<end-ip>.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 32 |
| Default value | N/A | Can be edited | Yes |

range_name

Deprecated, replaced by **dhcprange_name**.

start_addr

Deprecated, replaced by **dhcprange_start_addr**.

dhcprange_start_addr

The first IP address of the DHCPv4 range.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

end_addr

Deprecated, replaced by **dhcprange_end_addr**.

dhcprange_end_addr

The last IP address of the DHCPv4 range.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 23. DHCPv6 Range

Name

dhcp6_range6_add — Add/Edit a DHCPv6 range

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcprange6_start_addr && dhcprange6_end_addr && (dhcpscope6_id || dhcp6_id || dhcp6_name || hostaddr))
- **Editing:** (dhcprange6_id || (dhcprange6_start_addr && dhcprange6_end_addr && (dhcpscope6_id || dhcp6_id || dhcp6_name || hostaddr)))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpscope6_name

The name of the DHCPv6 scope.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope, a unique numeric key value automatically incremented when you add a DHCPv6 scope. Use the ID to specify the DHCPv6 scope of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcprange6_id

The database identifier (ID) of the DHCPv6 range, a unique numeric key value automatically incremented when you add a DHCPv6 range. Use the ID to specify the DHCPv6 range of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcprange6_start_addr

The first IP address of the DHCPv6 range.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcprange6_end_addr

The last IP address of the DHCPv6 range.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcprange6_acl

The list of ACLs associated with the DHCPv6 range, as follows: <ACL_name>;<ACL_name>;...

| | | | |
|---------------|--------|----------------|------|
| Type | String | Maximum length | 4000 |
| Default value | N/A | Can be edited | Yes |

dhcprange6_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

dhcprange6_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcprange6_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - *set, propagate* - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp6_range6_list — List the DHCPv6 ranges

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcprange6_id

The database identifier (ID) of the DHCPv6 range.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope the object belongs to.

dhcprange6_start_ip6_addr

The first IP address of the DHCPv6 range, in hexadecimal format.

dhcprange6_end_ip6_addr

The last IP address of the DHCPv6 range, in hexadecimal format.

dhcprange6_name

The start and end IP address of the DHCPv6 range, **dhcprange6_start_ip6_addr** and **dhcprange6_end_ip6_addr**, in compressed format as follows: <start-ip> - <end-ip>.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 23.1. *dhcp6_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

dhcp6_name

The name of the DHCPv6 server the object belongs to.

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpscope6_name

The name of the DHCPv6 scope the object belongs to.

dhcpscope6_start_ip6_addr

The first IP address of the DHCPv6 scope the object belongs to, in hexadecimal format.

dhcpscope6_end_ip6_addr

The last IP address of the DHCPv6 scope the object belongs to, in hexadecimal format.

dhcpscope6_size

The number of IP addresses the DHCPv6 scope the object belongs to contains.

dhcpscope6_prefix

The prefix of the DHCPv6 scope the object belongs to.

dhcpscope6_site_id

The database identifier (ID) of the space associated with the DHCPv6 scope the object belongs to.

dhcprange6_failover_name

Internal use. Not documented.

dhcprange6_state

Internal use. Not documented.

dhcprange6_class_name

The name of the class applied to the DHCPv6 range, it can be preceded by the class directory.

dhcprange6_lease_count

The total number of leases currently delivered by the DHCPv6 range.

dhcprange6_size

The number of IP addresses the DHCPv6 range contains.

dhcprange6_acl

The list of ACLs associated with the DHCPv6 range, as follows: <ACL_name>;<ACL_name>;...

dhcpsn6_id

The database identifier (ID) of the DHCPv6 shared network the object belongs to.

dhcpsn6_name

The name of the DHCPv6 shared network the object belongs to.

vdhcp6_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp6_comment

The description of the DHCPv6 server the object belongs to.

dhcpscope6_class_name

The name of the class applied to the DHCPv6 scope the object belongs to, it can be preceded by the class directory.

dhcp6_class_name

The name of the class applied to the DHCPv6 server the object belongs to, it can be preceded by the class directory.

dhcp6_version

The version details of the DHCPv6 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

ip_addr

The Management IP address of the DHCPv6 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv6 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number> @<multi-status-severity> @<module>. The different severity levels are:

Table 23.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcprange6_class_parameters

The class parameters applied to the DHCPv6 range and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcprange6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcprange6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcprange6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcpscope6_class_parameters

The class parameters applied to the DHCPv6 scope the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpscope6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcp6_class_parameters

The class parameters applied to the DHCPv6 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

Name

dhcp6_range6_info — Display the properties of a DHCPv6 range

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcprange6_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

dhcprange6_id

The database identifier (ID) of the DHCPv6 range, a unique numeric key value automatically incremented when you add a DHCPv6 range. Use the ID to specify the DHCPv6 of your choice.

Output Parameters

dhcprange6_id

The database identifier (ID) of the DHCPv6 range.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope the object belongs to.

dhcprange6_start_ip6_addr

The first IP address of the DHCPv6 range, in hexadecimal format.

dhcprange6_end_ip6_addr

The last IP address of the DHCPv6 range, in hexadecimal format.

dhcprange6_name

The start and end IP address of the DHCPv6 range, **dhcprange6_start_ip6_addr** and **dhcprange6_end_ip6_addr**, in compressed format as follows: <start-ip> - <end-ip>.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 23.3. *dhcp6_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

dhcp6_name

The name of the DHCPv6 server the object belongs to.

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpscope6_name

The name of the DHCPv6 scope the object belongs to.

dhcpscope6_start_ip6_addr

The first IP address of the DHCPv6 scope the object belongs to, in hexadecimal format.

dhcpscope6_end_ip6_addr

The last IP address of the DHCPv6 scope the object belongs to, in hexadecimal format.

dhcpscope6_size

The number of IP addresses the DHCPv6 scope the object belongs to contains.

dhcpscope6_prefix

The prefix of the DHCPv6 scope the object belongs to.

dhcpscope6_site_id

The database identifier (ID) of the space associated with the DHCPv6 scope the object belongs to.

dhcprange6_failover_name

Internal use. Not documented.

dhcprange6_state

Internal use. Not documented.

dhcprange6_class_name

The name of the class applied to the DHCPv6 range, it can be preceded by the class directory.

dhcprange6_lease_count

The total number of leases currently delivered by the DHCPv6 range.

dhcprange6_size

The number of IP addresses the DHCPv6 range contains.

dhcprange6_acl

The list of ACLs associated with the DHCPv6 range, as follows: <ACL_name>;<ACL_name>;...

dhcpsn6_id

The database identifier (ID) of the DHCIPv6 shared network the object belongs to.

dhcpsn6_name

The name of the DHCIPv6 shared network the object belongs to.

vdhcp6_parent_name

The name of the DHCIPv4 smart architecture managing the DHCIPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcp6_comment

The description of the DHCIPv6 server the object belongs to.

dhcpscope6_class_name

The name of the class applied to the DHCIPv6 scope the object belongs to, it can be preceded by the class directory.

dhcp6_class_name

The name of the class applied to the DHCIPv6 server the object belongs to, it can be preceded by the class directory.

dhcp6_version

The version details of the DHCIPv6 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

ip_addr

The Management IP address of the DHCIPv6 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCIPv6 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 23.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|---|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcprange6_class_parameters

The class parameters applied to the DHCPv6 range and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcprange6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcprange6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcprange6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcpscope6_class_parameters

The class parameters applied to the DHCPv6 scope the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpscope6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcp6_class_parameters

The class parameters applied to the DHCPv6 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dhcp6_range6_count — Count the number of DHCPv6 ranges

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp6_range6_options6_list — List the DHCP options set on a DHCPv6 range

Description

This service allows you to list the DHCP options set on a specific range. To add, edit or delete DHCP options, refer to the chapter [DHCPv6 Option](#).

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcprange6_id

The database identifier (ID) of the DHCPv6 range, a unique numeric key value automatically incremented when you add a DHCPv6 range. Use the ID to specify the DHCPv6 of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>' or <parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcprangeoption6_id

The database identifier (ID) of the DHCP option set on the DHCPv6 range.

oid

Internal use. Not documented.

row_enabled

The object activation status:

- If set to *0*, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to *1*, the object is enabled and managed.

By default, *row_enabled* is set to *1* when an object is created.

dhcprange6_id

The database identifier (ID) of the DHCPv6 range.

dhcpoption6_name

The name of the DHCPv6 option.

dhcpoption6_value

The value of the DHCPv6 option.

delayed_time

The delay of creation/deletion status. *1* indicates that the object is not created/deleted yet.

modif_count

Internal use. Not documented.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 23.5. dhcp6_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

Name

dhcp6_range6_delete — Delete a DHCPv6 range

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcprange6_id || ((dhcprange6_start_addr || dhcprange6_end_addr) && (dhcp6_id || dhcp6_name || hostaddr || dhcpscope6_id)))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope, a unique numeric key value automatically incremented when you add a DHCPv6 scope. Use the ID to specify the DHCPv6 scope of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcprange6_id

The database identifier (ID) of the DHCPv6 range, a unique numeric key value automatically incremented when you add a DHCPv6 range. Use the ID to specify the DHCPv6 of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|------|-------------|----------------|-----|
|------|-------------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

dhcprange6_start_addr

The first IP address of the DHCPv6 range.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcprange6_end_addr

The last IP address of the DHCPv6 range.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 24. DHCPv4 Lease

Name

dhcp_range_lease_list — List the DHCPv4 leases

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcplease_giaddr

The gateway IP address of the relay agent of the DHCPv4 lease.

dhcplease_vendor_id

The vendor class identifier (ID) of the client associated with the DHCPv4 lease.

dhcplease_fingerbank_os

The operating system details of the client associated with the DHCPv4 lease.

dhcplease_remote_id

The remote identifier (ID) of the relay agent associated with the DHCPv4 lease.

dhcplease_circuit_id

The circuit identifier (ID) of the relay agent associated with the DHCPv4 lease.

parameter_request_list

The list of parameters requested with the DHCPv4 lease returned by the server, integers separated by a comma.

mac_vendor

The vendor details of the client associated with the DHCPv4 lease.

dhcplease_id

The database identifier (ID) of the DHCPv4 lease.

dhcplease_addr

The IP address associated with the DHCPv4 lease.

dhcplease_ip_addr

The IP address associated with the DHCPv4 lease, in hexadecimal format.

dhcplease_mac_addr

The MAC address associated with the IPv4 lease.

dhcplease_client_ident

The client identifier (ID) of the client associated with the DHCPv4 lease.

dhcplease_first_time

The first time the DHCPv4 lease has been attributed to the client, in decimal UNIX date format.

dhcplease_time

The last time the DHCPv4 lease has been attributed to the client, in decimal UNIX date format.

dhcplease_end_time

The expiration time of the lease, in decimal UNIX date format.

dhcplease_period

The duration time (time to live) of the DHCPv4 lease, in seconds.

percent

The percentage of time the lease has really been in use.

time_to_expire

The time left to the lease before it expires, in seconds.

dhcplease_name

The name of the DHCPv4 lease.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope the object belongs to.

dhcprange_id

The database identifier (ID) of the DHCPv4 range the object belongs to.

dhcplease_domain

The domain name associated with the DHCPv4 lease.

dhcprange_name

The start and end IP address of the DHCPv4 range the object belongs to, **dhcprange_start_addr** and **dhcprange_end_addr**, as follows: <start-ip>-<end-ip>.

dhcprange_start_addr

The first IP address of the DHCPv4 range the lease belongs to.

dhcprange_end_addr

The last IP address of the DHCPv4 range the lease belongs to.

dhcpscope_name

The name of the DHCPv4 scope the object belongs to.

dhcpscope_size

The number of IP addresses the DHCPv4 scope the object belongs to contains.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope the object belongs to.

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network the object belongs to.

dhcpsn_name

The name of the DHCPv4 shared network the object belongs to.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 24.1. *dhcp_type* possible values

| Type | Description |
|------|---|
| ipm | EfficientIP or EfficientIP Package server |

| Type | Description |
|-------|---------------------------------------|
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active (M)*, *passive (B)* or *N/A (#)*.

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcprange_failover_name

Internal use. Not documented.

dhcprange_class_name

The name of the class applied to the DHCPv4 range the object belongs to, it can be preceded by the class directory.

dhcpscope_class_name

The name of the class applied to the DHCPv4 scope the object belongs to, it can be preceded by the class directory.

dhcp_class_name

The name of the class applied to the DHCPv4 server the object belongs to, it can be preceded by the class directory.

dhcp_version

The version details of the DHCPv4 server the object belongs to.

ip_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 24.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|---|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcprange_class_parameters

The class parameters applied to the DHCPv4 range the object belongs to and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&....

dhcprange_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcprange_class_parameters: *<class-parameter1>=<inheritance><propagation>&<class-parameter2>=<inheritance>&....*

dhcprange_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:
<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&....

Example

In the example below, we call the service **dhcp_range_lease_list** with Python (Requests) using the clause *WHERE* to list all the leases associated with a specific MAC address that have a time to expire inferior or equal to an hour, or 3600 seconds.

Example 24.1. Calling the service dhcp_range_lease_list using Python and WHERE

```
import requests

url = "https://solid.intranet/rest/dhcp_range_lease_list"

querystring = {"WHERE": "dhcplease_mac_addr='01:08:00:27:d9:4e:28' and time_to_expire<=3600"}

headers = {
    'x-ipm-username': "aXBtYWRtaW4=",
    'x-ipm-password': "YWRtaW4=",
    'cache-control': "no-cache"
}

response = requests.request("GET", url, headers=headers, params=querystring)

print(response.text)
```

Name

dhcp_range_lease_info — Display the properties of a DHCPv4 lease

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcplease_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

dhcplease_id

The database identifier (ID) of the DHCPv4 lease, a unique numeric key value automatically incremented when you add a DHCPv4 lease. Use the ID to specify the DHCPv4 lease of your choice.

Output Parameters

dhcplease_giaddr

The gateway IP address of the relay agent of the DHCPv4 lease.

dhcplease_vendor_id

The vendor class identifier (ID) of the client associated with the DHCPv4 lease.

dhcplease_fingerbank_os

The operating system details of the client associated with the DHCPv4 lease.

dhcplease_remote_id

The remote identifier (ID) of the relay agent associated with the DHCPv4 lease.

dhcplease_circuit_id

The circuit identifier (ID) of the relay agent associated with the DHCPv4 lease.

parameter_request_list

The list of parameters requested with the DHCPv4 lease returned by the server, integers separated by a comma.

mac_vendor

The vendor details of the client associated with the DHCPv4 lease.

dhcplease_id

The database identifier (ID) of the DHCPv4 lease.

dhcplease_addr

The IP address associated with the DHCPv4 lease.

dhcplease_ip_addr

The IP address associated with the DHCPv4 lease, in hexadecimal format.

dhcplease_mac_addr

The MAC address associated with the IPv4 lease.

dhcplease_client_ident

The client identifier (ID) of the client associated with the DHCPv4 lease.

dhcplease_first_time

The first time the DHCPv4 lease has been attributed to the client, in decimal UNIX date format.

dhcplease_time

The last time the DHCPv4 lease has been attributed to the client, in decimal UNIX date format.

dhcplease_end_time

The expiration time of the lease, in decimal UNIX date format.

dhcplease_period

The duration time (time to live) of the DHCPv4 lease, in seconds.

percent

The percentage of time the lease has really been in use.

time_to_expire

The time left to the lease before it expires, in seconds.

dhcplease_name

The name of the DHCPv4 lease.

dhcplease_clientname

The name of the client associated with the DHCPv4 lease.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope the object belongs to.

dhcprange_id

The database identifier (ID) of the DHCPv4 range the object belongs to.

dhcplease_domain

The domain name associated with the DHCPv4 lease.

dhcprange_name

The start and end IP address of the DHCPv4 range the object belongs to, **dhcprange_start_addr** and **dhcprange_end_addr**, as follows: <start-ip>-<end-ip>.

dhcprange_start_addr

The first IP address of the DHCPv4 range the lease belongs to.

dhcprange_end_addr

The last IP address of the DHCPv4 range the lease belongs to.

dhcpscope_name

The name of the DHCPv4 scope the object belongs to.

dhcpscope_size

The number of IP addresses the DHCPv4 scope the object belongs to contains.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope the object belongs to.

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network the object belongs to.

dhcpsn_name

The name of the DHCPv4 shared network the object belongs to.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 24.3. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active (M)*, *passive (B)* or *N/A (#)*.

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcprange_failover_name

Internal use. Not documented.

dhcprange_class_name

The name of the class applied to the DHCPv4 range the object belongs to, it can be preceded by the class directory.

dhcpscope_class_name

The name of the class applied to the DHCPv4 scope the object belongs to, it can be preceded by the class directory.

dhcp_class_name

The name of the class applied to the DHCPv4 server the object belongs to, it can be preceded by the class directory.

dhcp_version

The version details of the DHCPv4 server the object belongs to.

ip_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 24.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcprange_class_parameters

The class parameters applied to the DHCPv4 range the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcprange_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcprange_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcprange_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

dhcp_range_lease_count — Count the number of DHCPv4 leases

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp_range_lease_groupby — Group DHCPv4 leases by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

dhcp_range_lease_groupby_count — Count the number of DHCPv4 leases grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

dhcp_lease_log_list — List the DHCPv4 leases moved to the *Lease history*

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 24.5. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active* (*M*), *passive* (*B*) or *N/A* (*#*).

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

histo_time

The time the DHCPv4 lease has been attributed to the client, in decimal UNIX date format.

histo_last_time

The expiration time of the DHCPv4 lease, in decimal UNIX date format.

dhcplease_period

The duration time (time to live) of the DHCPv4 lease, in seconds.

histo_state

Internal use. Not documented.

dhcplease_addr

The IP address associated with the DHCPv4 lease.

dhcplease_ip_addr

The IP address associated with the DHCPv4 lease, in hexadecimal format.

mac_addr

The MAC address associated with the IPv4 lease.

domain

The domain name associated with the DHCPv4 lease.

name

The name of the DHCPv4 lease.

dhcplease_full_name

The full name of the DHCPv4 lease in FQDN format: <lease-name>-<domain-name>.

dhcplease_id

The database identifier (ID) of the DHCPv4 lease.

dhcplease_histo_id

The database identifier (ID) of the lease log.

client_id

The client identifier (ID) of the client associated with the DHCPv4 lease.

remote_id

The remote identifier (ID) of the relay agent associated with the DHCPv4 lease.

circuit_id

The circuit identifier (ID) of the relay agent associated with the DHCPv4 lease.

parameter_request_list

The list of parameters requested with the DHCPv4 lease returned by the server, integers separated by a comma.

dhcplease_fingerbank_os

The operating system details of the client associated with the DHCPv4 lease.

dhcplease_giaddr

The gateway IP address of the relay agent of the DHCPv4 lease.

dhcpscope_name

The name of the DHCPv4 scope the object belongs to.

dhcprange_name

The start and end IP address of the DHCPv4 range the object belongs to, as follows: <*start-ip*>-<*end-ip*>.

Name

dhcp_lease_log_count — Count the number of DHCPv4 leases moved to the *Lease history*

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause *WHERE*.

Name

dhcp_lease_log_groupby — Group DHCPv4 leases moved to the *Lease history* by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. If you specify several parameters they must be separated by a comma as follows: `GROUPBY=<param1>,<param2>,...`. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter `offset` must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter `limit` must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement `SELECT` is returned.

Name

dhcp_lease_log_groupby_count — Count the number of DHCPv4 leases moved to the *Lease history* grouped by parameter(s)

Description

This service allows you to display the total number of results of the service **_groupby*.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

dhcp_lease_manual_delete — Release a DHCPv4 lease

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcplease_id || (dhcplease_addr && (dhcp_id || dhcp_name)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcplease_id

The database identifier (ID) of the DHCPv4 lease, a unique numeric key value automatically incremented when you add a DHCPv4 lease. Use the ID to specify the DHCPv4 lease of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcplease_addr

The IP address associated with the DHCPv4 lease.

| Type | IPv4 address | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

lease_addr

Deprecated, replaced by **dhcplease_addr**.

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 25. DHCPv6 Lease

Name

dhcp6_lease6_list — List the DHCPv6 leases

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

mac_vendor

The vendor details of the client associated with the DHCPv6 lease.

dhcplease6_id

The database identifier (ID) of the DHCPv6 lease.

dhcplease6_ip6_addr

The IP address associated with the DHCPv6 lease, in hexadecimal format.

dhcplease6_giaddr

The gateway IP address of the relay agent of the DHCPv6 lease.

dhcplease6_mac_addr

The MAC address associated with the DHCPv6 lease.

dhcplease6_client_duid

The client DHCP Unique Identifier (DUID) associated with the DHCPv6 lease.

dhcplease6_first_time

The first time the DHCPv6 lease has been attributed to the client, in decimal UNIX date format.

dhcplease6_time

The last time the DHCPv6 lease has been attributed to the client, in decimal UNIX date format.

dhcplease6_end_time

The expiration time of the lease, in decimal UNIX date format.

dhcplease6_period

The duration time (time to live) of the DHCPv6 lease, in seconds.

percent

The percentage of time the lease has really been in use.

time_to_expire

The time left to the lease before it expires, in seconds.

dhcplease6_name

The name of the DHCPv6 lease.

dhcplease6_clientname

The name of the client associated with the DHCPv6 lease.

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope the object belongs to.

dhcprange6_id

The database identifier (ID) of the DHCPv6 range the object belongs to.

dhcplease6_domain

The domain name associated with the DHCPv6 lease.

dhcprange6_start_ip6_addr

The first IP address of the DHCPv6 range the object belongs to, in hexadecimal format.

dhcprange6_end_ip6_addr

The last IP address of the DHCPv6 range the object belongs to, in hexadecimal format.

dhcpscope6_name

The name of the DHCPv6 scope the object belongs to.

dhcpscope6_size

The number of IP addresses the DHCPv6 scope the object belongs to contains.

dhcpscope6_start_ip6_addr

The first IP address of the DHCPv6 scope the object belongs to, in hexadecimal format.

dhcpscope6_prefix

The prefix of the DHCPv6 scope the object belongs to.

dhcpsn6_id

The database identifier (ID) of the DHCPv6 shared network the object belongs to.

dhcpsn6_name

The name of the DHCPv6 shared network the object belongs to.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 25.1. *dhcp6_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

vdhcp6_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcprange6_failover_name

Internal use. Not documented.

dhcprange6_class_name

The name of the class applied to the DHCPv6 range the object belongs to, it can be preceded by the class directory.

dhcpscope6_class_name

The name of the class applied to the DHCIPv6 scope the object belongs to, it can be preceded by the class directory.

dhcp6_class_name

The name of the class applied to the DHCIPv6 server the object belongs to, it can be preceded by the class directory.

dhcp6_version

The version details of the DHCIPv6 server the object belongs to.

ip_addr

The Management IP address of the DHCIPv6 server the object belongs to, the IPv4 address configured when adding the server, in hexadecimal format.

ip6_addr

The Management IP address of the DHCIPv6 server the object belongs to, the IPv6 address configured when adding the server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number> @<multi-status-severity> @<module>. The different severity levels are:

Table 25.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcprange6_class_parameters

The class parameters applied to the DHCIPv6 range the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcprange6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcprange6_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcprange6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

dhcp6_lease6_count — Count the number of DHCPv6 leases

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp6_lease6_log_list — List the DHCPv6 leases moved to the *Lease history*

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter **offset** must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter **limit** must be specified in **lowercase**.

Output Parameters

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 25.3. *dhcp6_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

histo_time

The time the DHCPv6 lease has been attributed to the client, in decimal UNIX date format.

histo_last_time

The expiration time of the DHCPv6 lease, in decimal UNIX date format.

dhcplease6_period

The duration time (time to live) of the DHCPv6 lease, in seconds.

histo_state

Internal use. Not documented.

dhcplease6_ip6_addr

The IP address associated with the DHCPv6 lease, in hexadecimal format.

mac_addr

The MAC address associated with the DHCPv6 lease.

domain

The domain name associated with the DHCPv6 lease.

name

The name of the DHCPv6 lease.

dhcplease6_full_name

The full name of the DHCPv6 lease, as follows: <dhcplease6_name>.<dhcplease6_domain>

dhcplease6_id

The database identifier (ID) of the DHCPv6 lease.

dhcplease6_histo_id

The database identifier (ID) of the DHCPv6 lease log.

client_duid

The client DHCP Unique Identifier (DUID) associated with the DHCPv6 lease.

dhcplease6_giaddr

The gateway IP address of the relay agent of the DHCPv6 lease.

dhcpscope6_name

The name of the DHCPv6 scope the object belongs to.

dhcprange6_start_ip6_addr

The first IP address of the DHCPv6 range the object belongs to, in hexadecimal format.

dhcprange6_end_ip6_addr

The last IP address of the DHCPv6 range the object belongs to, in hexadecimal format.

Name

dhcp6_lease6_log_count — List the DHCPv6 leases moved to the *Lease history*

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause *WHERE*.

Name

dhcp6_lease6_log_groupby — Group DHCPv6 leases moved to the *Lease history* by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. If you specify several parameters they must be separated by a comma as follows: `GROUPBY=<param1>,<param2>,...`. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter `offset` must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter `limit` must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement `SELECT` is returned.

Name

dhcp6_lease6_log_groupby_count — Count the number of DHCPv6 leases moved to the *Lease history* grouped by parameter(s)

Description

This service allows you to display the total number of results of the service **_groupby*.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Chapter 26. DHCPv4 Static

Name

dhcp_static_add — Add/Edit a DHCPv4 static

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcp_id || dhcp_name || hostaddr) && (dhcphost_addr || dhcphost_mac_addr || dhcphost_identifier)
- **Editing:** (dhcphost_id || (dhcp_id || dhcp_name || hostaddr) && (dhcphost_addr || dhcphost_mac_addr || dhcphost_identifier))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

static_id

Deprecated, replaced by **dhcphost_id**.

dhcphost_id

The database identifier (ID) of the DHCPv4 static, a unique numeric key value automatically incremented when you add a DHCPv4 static. Use the ID to specify the DHCPv4 static of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

static_ip_addr

Deprecated, replaced by **dhcphost_addr**.

static_addr

Deprecated, replaced by **dhcphost_addr**.

dhcphost_addr

The IP address associated with the DHCPv4 static.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | 0.0.0.0 | Can be edited | Yes |

dhcphost_name

The name of the DHCPv4 static, each DHCPv4 static must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

static_name

Deprecated, replaced by **dhcphost_name**.

static_mac_addr

Deprecated, replaced by **dhcphost_mac_addr**.

dhcphost_mac_addr

The MAC address you want to associate with the IPv4 static, it must include the MAC address type. The address has 7 sections, *00:11:22:33:44:55:66*, where *00* indicates the type. For Ethernet, type in *01*.

| | | | |
|---------------|-------------|----------------|-----|
| Type | MAC address | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dhcpgroup_id

The database identifier (ID) of the DHCPv4 group, a unique numeric key value automatically incremented when you add a DHCPv4 group. Use the ID to specify the DHCPv4 group of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dhcpgroup_name

The name of the DHCPv4 group.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

static_host_identifier

Deprecated, replaced by **dhcphost_identifier**.

dhcphost_identifier

The host identifier you want to associate with the IPv4 static. An option and value to look for to identify clients and assign them the static, specified as follows: *option <option-name>* "expected value".

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dhcpstatic_class_name

Deprecated, replaced by **dhcphost_class_name**.

dhcphost_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

dhcpstatic_class_parameters

Deprecated, replaced by **dhcphost_class_parameters**.

dhcphost_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: *<class-parameter1>=<value1>&<class-parameter2>=<value2>&....*

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dhcpstatic_class_parameters_properties

Deprecated, replaced by **dhcphost_class_parameters_properties**.

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: *<class-parameter1>&<class-parameter2>&....*. Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcphost_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Example

In the example below, we call the service **dhcp_static_add** with Ruby (NET::Http) to add a DHCP static without IP in one of our DHCP servers.

Example 26.1. Calling the service dhcp_static_add using Ruby

```
require 'uri'
require 'net/http'

url = URI("https://solid.intranet/rest/dhcp_static_add?"+
          "dhcphost_mac_addr=01%3A0a%B92%3Bf2%3B54%3A17%3A60&dhcp_id=19")

http = Net::HTTP.new(url.host, url.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE

request = Net::HTTP::Post.new(url)
request["x-ipm-username"] = 'aXBtYWRtaW4='
request["x-ipm-password"] = 'YWRtaW4='
request["cache-control"] = 'no-cache'

response = http.request(request)
puts response.read_body
```

Name

dhcp_static_list — List the DHCPv4 statics

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcphost_last_seen

The last time the MAC address associated with the DHCPv4 static was seen on the network, in decimal UNIX date format.

dhcphost_expire_time

The expiration time of the lease associated with the DHCPv4 static, in decimal UNIX date format.

dhcpscope_row_enabled

Internal use. Not documented.

dhcpscope_start_ip_addr

The first IP address of the DHCPv4 scope the object belongs to, in hexadecimal format.

dhcpscope_end_ip_addr

The last IP address of the DHCPv4 scope the object belongs to, in hexadecimal format.

dhcpscope_net_mask

The netmask of the DHCPv4 scope the object belongs to. It is expressed in dot-decimal notation and defines the number of addresses the scope contains.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope the object belongs.

mac_vendor

The vendor details of the client associated with the DHCPv4 static.

dhcphost_id

The database identifier (ID) of the DHCPv4 static.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcphost_mac_addr

The MAC address associated with the DHCPv4 static. It is composed of 7 sections, *00:11:22:33:44:55:66*, where *00* is the MAC address type. The type *01* indicates Ethernet.

dhcphost_addr

The IP address associated with the DHCPv4 static.

dhcphost_ip_addr

The IP address associated with the DHCPv4 static, in hexadecimal format.

dhcphost_identifier

The host identifier of the DHCPv4 static, specified as follows: *option <option-name> "expected value".*

db_hostname

Internal use. Not documented.

dhcphost_name

The name of the DHCPv4 static.

dhcphost_domain

The domain name associated with the DHCPv4 static.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 26.1. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcphost_state

Internal use. Not documented.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope the object belongs to.

dhcpscope_name

The name of the DHCPv4 scope the object belongs to.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

dhcpscope_size

The number of IP addresses the DHCPv4 scope the object belongs to contains.

dhcpscope_site_id

The database identifier (ID) of the space associated with the DHCPv4 scope the object belongs to.

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network the object belongs to.

dhcpsn_name

The name of the DHCPv4 shared network the object belongs to.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

dhcpgroup_id

The database identifier (ID) of the DHCPv4 group.

dhcpgroup_name

The name of the DHCPv4 group associated with the object.

dhcpgroup_class_name

The name of the class applied to the DHCPv4 group the static belongs to, it can be preceded by the class directory.

dhcphost_class_name

The name of the class applied to the DHCPv4 static, it can be preceded by the class directory.

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcpscope_class_name

The name of the class applied to the DHCPv4 scope the object belongs to, it can be preceded by the class directory.

dhcp_class_name

The name of the class applied to the DHCPv4 server the object belongs to, it can be preceded by the class directory.

dhcp_version

The version details of the DHCPv4 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 26.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcphost_class_parameters

The class parameters applied to the DHCPv4 static and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcphost_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcphost_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcphost_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:
<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcpgroup_class_parameters

The class parameters applied to the DHCPv4 group the static belongs to, and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpgroup_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcpgroup_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcpscope_class_parameters

The class parameters applied to the DHCPv4 scope the object belongs to and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcpscope_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcp_class_parameters

The class parameters applied to the DHCPv4 server the object belongs to and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcp_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dhcp_static_info — Display the properties of a DHCPv4 static

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpstatic_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

dhcpstatic_id

The database identifier (ID) of the DHCPv4 static, a unique numeric key value automatically incremented when you add a DHCPv4 static. Use the ID to specify the DHCPv4 static of your choice.

Output Parameters

dhcphost_last_seen

The last time the MAC address associated with the DHCPv4 static was seen on the network, in decimal UNIX date format.

dhcphost_expire_time

The expiration time of the lease associated with the DHCPv4 static, in decimal UNIX date format.

dhcpscope_row_enabled

Internal use. Not documented.

dhcpscope_start_ip_addr

The first IP address of the DHCPv4 scope the object belongs to, in hexadecimal format.

dhcpscope_end_ip_addr

The last IP address of the DHCPv4 scope the object belongs to, in hexadecimal format.

dhcpscope_net_mask

The netmask of the DHCPv4 scope the object belongs to. It is expressed in dot-decimal notation and defines the number of addresses the scope contains.

dhcpscope_net_addr

The first IP address of the DHCPv4 scope the object belongs.

mac_vendor

The vendor details of the client associated with the DHCPv4 static.

dhcphost_id

The database identifier (ID) of the DHCPv4 static.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcphost_mac_addr

The MAC address associated with the DHCPv4 static. It is composed of 7 sections, *00:11:22:33:44:55:66*, where *00* is the MAC address type. The type *01* indicates Ethernet.

dhcphost_addr

The IP address associated with the DHCPv4 static.

dhcphost_ip_addr

The IP address associated with the DHCPv4 static, in hexadecimal format.

dhcphost_identifier

The host identifier of the DHCPv4 static, specified as follows: *option <option-name> "expected value"*.

db_hostname

Internal use. Not documented.

dhcphost_name

The name of the DHCPv4 static.

dhcphost_domain

The domain name associated with the DHCPv4 static.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 26.3. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. *0* indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcphost_state

Internal use. Not documented.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope the object belongs to.

dhcpscope_name

The name of the DHCPv4 scope the object belongs to.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

dhcpscope_size

The number of IP addresses the DHCPv4 scope the object belongs to contains.

dhcpscope_site_id

The database identifier (ID) of the space associated with the DHCPv4 scope the object belongs to.

dhcpsn_id

The database identifier (ID) of the DHCPv4 shared network the object belongs to.

dhcpsn_name

The name of the DHCPv4 shared network the object belongs to.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

dhcpgroup_id

The database identifier (ID) of the DHCPv4 group.

dhcpgroup_name

The name of the DHCPv4 group associated with the object.

dhcpgroup_class_name

The name of the class applied to the DHCPv4 group the static belongs to, it can be preceded by the class directory.

dhcphost_class_name

The name of the class applied to the DHCPv4 static, it can be preceded by the class directory.

vdhcp_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. #indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcpscope_class_name

The name of the class applied to the DHCPv4 scope the object belongs to, it can be preceded by the class directory.

dhcp_class_name

The name of the class applied to the DHCPv4 server the object belongs to, it can be preceded by the class directory.

dhcp_version

The version details of the DHCPv4 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 26.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcphost_class_parameters

The class parameters applied to the DHCPv4 static and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcphost_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcphost_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcphost_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcpgroup_class_parameters

The class parameters applied to the DHCPv4 group the static belongs to, and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpgroup_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpgroup_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcpscope_class_parameters

The class parameters applied to the DHCPv4 scope the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpscope_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dhcp_class_parameters

The class parameters applied to the DHCPv4 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Example

In the example below, we call the service **dhcp_static_info** with Python (Requests) to retrieve the properties of a specific DHCP static.

Example 26.2. Calling the service dhcp_static_info using Python

```
import requests

url = "https://solid.intranet/rest/dhcp_static_info"

querystring = {"dhcpstatic_id": "121"}

headers = {
    'x-ipm-username': "aXBtYWRtaW4=",
    'x-ipm-password': "YWRtaW4=",
    'cache-control': "no-cache"
}

response = requests.request("GET", url, headers=headers, params=querystring)

print(response.text)
```

Name

dhcp_static_count — Count the number of DHCPv4 statics

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp_static_groupby — Group DHCPv4 statics by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

dhcp_static_groupby_count — Count the number of DHCPv4 statics grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

dhcp_static_options_list — List the DHCP options set on a DHCPv4 static

Description

This service allows you to list the DHCP options set on a specific static. To add, edit or delete DHCP options, refer to the chapter [DHCPv4 Option](#).

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpstatic_id

The database identifier (ID) of the DHCPv4 static, a unique numeric key value automatically incremented when you add a DHCPv4 static. Use the ID to specify the DHCPv4 static of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>' or <parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcphostoption_id

The database identifier (ID) of the DHCP options set on the DHCPv4 static.

oid

Internal use. Not documented.

row_enabled

The object activation status:

- If set to *0*, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to *1*, the object is enabled and managed.

By default, *row_enabled* is set to *1* when an object is created.

dhcphost_id

The database identifier (ID) of the DHCPv4 static.

dhcption_name

The name of the DHCPv4 option.

dhcption_value

The value of the DHCPv4 option.

delayed_create_time

The delay of creation status. *1* indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. *1* indicates that the object is not deleted yet.

modif_count

Internal use. Not documented.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 26.5. dhcp_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

Name

dhcp_static_delete — Delete a DHCPv4 static

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcpghost_id || (dhcpghost_addr && (dhcpghost_mac_addr || dhcpghost_identifier) && (dhcp_id || dhcp_name || hostaddr || dhcpscope_id)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope, a unique numeric key value automatically incremented when you add a DHCPv4 scope. Use the ID to specify the DHCPv4 scope of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

scope_id

Deprecated, replaced by **dhcpscope_id**.

dhcphost_id

The database identifier (ID) of the DHCPv4 static, a unique numeric key value automatically incremented when you add a DHCPv4 static. Use the ID to specify the DHCPv4 static of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

static_id

Deprecated, replaced by **dhcphost_id**.

dhcphost_addr

The IP address associated with the DHCPv4 static.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

static_addr

Deprecated, replaced by **dhcphost_addr**.

static_ip_addr

Deprecated, replaced by **dhcphost_addr**.

dhcphost_mac_addr

The MAC address associated with the IPv4 static, it must include the MAC address type. The address has 7 sections, *00:11:22:33:44:55:66*, where *00* indicates the type. For Ethernet, type in *01*.

| | | | |
|---------------|-------------|----------------|-----|
| Type | MAC address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

static_mac_addr

Deprecated, replaced by **dhcphost_mac_addr**.

dhcphost_name

The name of the DHCPv4 static.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcphost_identifier

The host identifier associated with the IPv4 static. An option and value to look for to identify clients and assign them the static, specified as follows: *option <option-name> "expected value"*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 27. DHCPv6 Static

Name

dhcp6_static6_add — Add/Edit a DHCPv6 static

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcphost6_name && (dhcphost6_mac_addr || dhcphost6_client_duid) && (dhcp6_id || dhcp6_name || hostaddr))
- **Editing:** (dhcphost6_id || (dhcphost6_name && (dhcphost6_mac_addr || dhcphost6_client_duid) && (dhcp6_id || dhcp6_name || hostaddr)))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcphost6_id

The database identifier (ID) of the DHCPv6 static, a unique numeric key value automatically incremented when you add a DHCPv6 static. Use the ID to specify the DHCPv6 static of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcphost6_name

The name of the DHCPv6 static, each DHCPv6 static must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dhcphost6_addr

The IP address associated with the DHCPv6 static.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | # | Can be edited | Yes |

dhcphost6_mac_addr

The MAC address you want to associate with the IPv6 static.

| | | | |
|---------------|-------------|----------------|-----|
| Type | MAC address | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dhcphost6_client_duid

The client DHCP Unique Identifier (DUID) associated with the DHCPv6 static.

| | | | |
|---------------|-----------------------------------|----------------|-----|
| Type | Regular expression: [:a-fA-F0-9]+ | Maximum length | 390 |
| Default value | | Can be edited | Yes |

dhcphost6_prefix6

The IP address of the delegated prefix of the DHCPv6 static.

| | | | |
|---------------|----------|----------------|-----|
| Type | ipv6_net | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcphost6_prefix6_addr

The prefix of the delegated prefix of the DHCPv6 static.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dhcphost6_prefix6_prefix

The IP address and prefix of the delegated prefix of the DHCPv6 static. You must specify them as follows: <IPv6-address>/<prefix>.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dhcpgroup6_id

The database identifier (ID) of the DHCPv6 group, a unique numeric key value automatically incremented when you add a DHCPv6 group. Use the ID to specify the DHCPv6 group of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dhcpgroup6_name

The name of the DHCPv6 group.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcphost6_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

dhcpstatic6_class_name

Deprecated, replaced by **dhcphost6_class_name**.

dhcphost6_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dhcpstatic6_class_parameters

Deprecated, replaced by **dhcphost6_class_parameters**.

dhcpstatic6_class_parameters_properties

Deprecated, replaced by **dhcphost6_class_parameters_properties**.

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcphost6_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be sep-

arated by a & and the properties by a comma: <class-parameter1>=<inheritance>, <propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp6_static6_list — List the DHCPv6 statics

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcphost6_client_duid

The client DHCP Unique Identifier (DUID) associated with the DHCPv6 static.

dhcphost6_time

Internal use. Not documented.

dhcphost6_end_time

Internal use. Not documented.

dhcpscope6_row_enabled

Internal use. Not documented.

dhcpscope6_start_ip6_addr

The first IP address of the DHCPv6 scope the object belongs to, in hexadecimal format.

dhcpscope6_end_ip6_addr

The last IP address of the DHCPv6 scope the object belongs to, in hexadecimal format.

mac_vendor

The vendor details of the client associated with the DHCPv6 static.

dhcphost6_id

The database identifier (ID) of the DHCPv6 static.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcphost6_mac_addr

The MAC address associated with the DHCPv6 static.

dhcphost6_ip6_addr

The IP address associated with the DHCPv6 static, in hexadecimal format.

dhcphost6_name

The name of the DHCPv6 static.

dhcphost6_domain

The domain name associated with the DHCPv6 static.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 27.1. dhcp6_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCIPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCIPv6 smart architecture managing the DHCIPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcphost6_state

Internal use. Not documented.

dhcpscope6_id

The database identifier (ID) of the DHCIPv6 scope the object belongs to.

dhcpscope6_name

The name of the DHCIPv6 scope the object belongs to.

dhcpscope6_size

The number of IP addresses the DHCIPv6 scope the object belongs to contains.

dhcpscope6_site_id

The database identifier (ID) of the space associated with the DHCIPv6 scope the object belongs to.

dhcpsn6_id

The database identifier (ID) of the DHCIPv6 shared network the object belongs to.

dhcpsn6_name

The name of the DHCIPv6 shared network the object belongs to.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

dhcpgroup6_id

The database identifier (ID) of the DHCIPv6 group.

dhcpgroup6_name

The name of the DHCIPv6 group associated with the object.

dhcpgroup6_class_name

The name of the class applied to the DHCIPv6 group the static belongs to, it can be preceded by the class directory.

dhcphost6_class_name

The name of the class applied to the DHCIPv6static, it can be preceded by the class directory.

vdhcp6_parent_name

The name of the DHCIPv4 smart architecture managing the DHCIPv4 server the object belongs to. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcpscope6_class_name

The name of the class applied to the DHCIPv6 scope the object belongs to, it can be preceded by the class directory.

dhcp6_class_name

The name of the class applied to the DHCIPv6 server the object belongs to, it can be preceded by the class directory.

dhcp6_version

The version details of the DHCPv6 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

dhcphost6_prefix6_addr

The IP address of the delegated prefix of the DHCPv6 static, in hexadecimal format.

dhcphost6_prefix6_prefix

The prefix of the delegated prefix of the DHCPv6 static.

dhcphost6_prefix6

The IP address and prefix of the delegated prefix of the DHCPv6 static, in hexadecimal format as follows: <IPv6-address>/<prefix>.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 27.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcphost6_class_parameters

The class parameters applied to the DHCPv6 static and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcphost6_class_parameters_properties

The DHCPv6 static class parameters inheritance property and propagation property, both encoded in URL format: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

If the inheritance or propagation property is not specified, its default value - *set*, *propagate* - is used.

dhcphost6_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcpgroup6_class_parameters

The class parameters applied to the DHCPv6 group the static belongs to, and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpgroup6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcpgroup6_class_parameters: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .*

dhcpscope6_class_parameters

The class parameters applied to the DHCPv6 scope the object belongs to and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcpscope6_class_parameters: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .*

dhcp6_class_parameters

The class parameters applied to the DHCPv6 server the object belongs to and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp6_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dhcp6_class_parameters: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .*

Name

dhcp6_static6_info — Display the properties of a DHCPv6 static

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

`dhcpstatic6_id`

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

dhcpstatic6_id

The database identifier (ID) of the DHCPv6 static, a unique numeric key value automatically incremented when you add a DHCPv6 static. Use the ID to specify the DHCPv6 static of your choice.

Output Parameters

dhcphost6_client_duid

The client DHCP Unique Identifier (DUID) associated with the DHCPv6 static.

dhcphost6_time

Internal use. Not documented.

dhcphost6_end_time

Internal use. Not documented.

dhcpscope6_row_enabled

Internal use. Not documented.

dhcpscope6_start_ip6_addr

The first IP address of the DHCPv6 scope the object belongs to, in hexadecimal format.

dhcpscope6_end_ip6_addr

The last IP address of the DHCPv6 scope the object belongs to, in hexadecimal format.

mac_vendor

The vendor details of the client associated with the DHCPv6 static.

dhcphost6_id

The database identifier (ID) of the DHCPv6 static.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcphost6_mac_addr

The MAC address associated with the DHCPv6 static.

dhcphost6_ip6_addr

The IP address associated with the DHCPv6 static, in hexadecimal format.

dhcphost6_name

The name of the DHCPv6 static.

dhcphost6_domain

The domain name associated with the DHCPv6 static.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 27.3. dhcp6_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcphost6_state

Internal use. Not documented.

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope the object belongs to.

dhcpscope6_name

The name of the DHCPv6 scope the object belongs to.

dhcpscope6_size

The number of IP addresses the DHCPv6 scope the object belongs to contains.

dhcpscope6_site_id

The database identifier (ID) of the space associated with the DHCPv6 scope the object belongs to.

dhcpsn6_id

The database identifier (ID) of the DHCPv6 shared network the object belongs to.

dhcpsn6_name

The name of the DHCPv6 shared network the object belongs to.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

dhcpgroup6_id

The database identifier (ID) of the DHCPv6 group.

dhcpgroup6_name

The name of the DHCPv6 group associated with the object.

dhcpgroup6_class_name

The name of the class applied to the DHCPv6 group the static belongs to, it can be preceded by the class directory.

dhcphost6_class_name

The name of the class applied to the DHCPv6static, it can be preceded by the class directory.

vdhcp6_parent_name

The name of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. #indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dhcpscope6_class_name

The name of the class applied to the DHCPv6 scope the object belongs to, it can be preceded by the class directory.

dhcp6_class_name

The name of the class applied to the DHCPv6 server the object belongs to, it can be preceded by the class directory.

dhcp6_version

The version details of the DHCPv6 server the object belongs to.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

dhcphost6_prefix6_addr

The IP address of the delegated prefix of the DHCPv6 static, in hexadecimal format.

dhcphost6_prefix6_prefix

The prefix of the delegated prefix of the DHCPv6 static.

dhcphost6_prefix6

The IP address and prefix of the delegated prefix of the DHCPv6 static, in hexadecimal format as follows: <IPv6-address>/<prefix>.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 27.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|---|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dhcp6_class_parameters

The class parameters applied to the DHCPv6 static and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp6_class_properties

The DHCPv6 static class parameters inheritance property and propagation property, both encoded in URL format: <class-parameter1>=<inheritance>, <propagation>&<class-parameter2>=<inheritance>&.... .

If the inheritance or propagation property is not specified, its default value - *set*, *propagate* - is used.

dhcp6_class_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dhcpgroup6_class_parameters

The class parameters applied to the DHCPv6 group the static belongs to, and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpgroup6_class_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpgroup6_class_parameters**: <class-parameter1>=<inheritance>, <propagation>&<class-parameter2>=<inheritance>&.... .

dhcpscope6_class_parameters

The class parameters applied to the DHCPv6 scope the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcpscope6_class_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcpscope6_class_parameters**: <class-parameter1>=<inheritance>, <propagation>&<class-parameter2>=<inheritance>&.... .

dhcp6_class_parameters

The class parameters applied to the DHCPv6 server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dhcp6_class_properties

The inheritance property and/or propagation property of the class parameters returned by **dhcp6_class_parameters**: <class-parameter1>=<inheritance>, <propagation>&<class-parameter2>=<inheritance>&.... .

Name

dhcp6_static6_count — Count the number of DHCPv6 statics

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp6_static6_options6_list — List the DHCP options set on a DHCPv6 static

Description

This service allows you to list the DHCP options set on a specific static. To add, edit or delete DHCP options, refer to the chapter [DHCPv4 Option](#).

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpstatic6_id

The database identifier (ID) of the DHCPv6 static, a unique numeric key value automatically incremented when you add a DHCPv6 static. Use the ID to specify the DHCPv6 static of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>' or <parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcphostoption6_id

The database identifier (ID) of the DHCP options set on the DHCPv6 static.

oid

Internal use. Not documented.

row_enabled

The object activation status:

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.

By default, *row_enabled* is set to 1 when an object is created.

dhcphost6_id

The database identifier (ID) of the DHCPv6 static.

dhcption6_name

The name of the DHCPv6 option.

dhcption6_value

The value of the DHCPv6 option.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

modif_count

Internal use. Not documented.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 27.5. *dhcp6_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. *0* indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

Name

dhcp6_static6_delete — Delete a DHCPv6 static

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcphost6_id || ((dhcphost6_name || dhcphost6_addr) && (dhcp6_id || dhcp6_name || hostaddr || dhcpscope6_id)))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope, a unique numeric key value automatically incremented when you add a DHCPv6 scope. Use the ID to specify the DHCPv6 scope of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcphost6_id

The database identifier (ID) of the DHCPv6 static, a unique numeric key value automatically incremented when you add a DHCPv6 static. Use the ID to specify the DHCPv6 static of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

dhcp6_name

The name of the DHCPv6 static.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dhcp6_addr

The IP address associated with the DHCPv6 static.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | IPv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 28. DHCPv4 Option

Name

dhcp_option_add — Add/Edit/Delete a DHCP option on DHCPv4 objects

Description

This service allows you to add, edit or delete DHCP options on all DHCP objects, except leases, failover channels and shared networks. The service `dhcp_option_delete` does not exist.

- If no identifier is specified, a new option is created.
- If an existing identifier is specified:
 - The value specified in input edits the option.
 - The option specified without value is deleted.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** `(dhcption_type && dhcption_name && dhcption_value && ((dhcp_id || dhcp_name) || (dhcpscope_id || (scope_name && (dhcp_id || dhcp_name || hostaddr))) || (dhcpacl_id || (acl_name && (dhcp_id || dhcp_name || hostaddr))) || (dhcpacl_data_id || (acl_data_value && (dhcp_id || dhcp_name || hostaddr))) || (dhcpgroup_id || (group_name && (dhcp_id || dhcp_name || hostaddr))) || (dhcprange_id || (range_name && (dhcp_id || dhcp_name || hostaddr))) || (dhcphost_id || (host_name && (dhcp_id || dhcp_name || hostaddr))))))`
- **Editing:** `(dhcption_type && dhcption_name && ((dhcp_id || dhcp_name) || (dhcpscope_id || (scope_name && (dhcp_id || dhcp_name || hostaddr))) || (dhcpacl_id || (acl_name && (dhcp_id || dhcp_name || hostaddr))) || (dhcpacl_data_id || (acl_data_value && (dhcp_id || dhcp_name || hostaddr))) || (dhcpgroup_id || (group_name && (dhcp_id || dhcp_name || hostaddr))) || (dhcprange_id || (range_name && (dhcp_id || dhcp_name || hostaddr))) || (dhcphost_id || (host_name && (dhcp_id || dhcp_name || hostaddr))))))`

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpscope_id

The database identifier (ID) of the DHCPv4 scope, a unique numeric key value automatically incremented when you add a DHCPv4 scope. Use the ID to specify the DHCPv4 scope of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

scope_id

Deprecated, replaced by **dhcpscope_id**.

dhcpscope_name

The name of the DHCPv4 scope.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

scope_name

Deprecated, replaced by **dhcpscope_name**.

dhcprange_id

The database identifier (ID) of the DHCPv4 range, a unique numeric key value automatically incremented when you add a DHCPv4 range. Use the ID to specify the DHCPv4 range of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

range_id

Deprecated, replaced by **dhcprange_id**.

dhcprange_name

The start and end IP address of the DHCPv4 range, as follows: <start-ip>-<end-ip>.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 32 |
| Default value | N/A | Can be edited | Yes |

range_name

Deprecated, replaced by **dhcprange_name**.

dhcphost_id

The database identifier (ID) of the DHCPv4 static, a unique numeric key value automatically incremented when you add a DHCPv4 static. Use the ID to specify the DHCPv4 static of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

host_id

Deprecated, replaced by **dhcphost_id**.

dhcphost_name

The name of the DHCPv4 static.

| | | | |
|------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
|------|--------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

host_name

Deprecated, replaced by **dhcphost_name**.

dhcpgroup_id

The database identifier (ID) of the DHCPv4 group, a unique numeric key value automatically incremented when you add a DHCPv4 group. Use the ID to specify the DHCPv4 group of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

group_id

Deprecated, replaced by **dhcpgroup_id**.

dhcpgroup_name

The name of the DHCPv4 group.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

group_name

Deprecated, replaced by **dhcpgroup_name**.

dhcpacl_id

The database identifier (ID) of the DHCPv4 ACL, a unique numeric key value automatically incremented when you add a DHCPv4 ACL. Use the ID to specify the DHCPv4 ACL of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

acl_id

Deprecated, replaced by **dhcpacl_id**.

dhcpacl_name

The name of the DHCPv4 ACL.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

acl_name

Deprecated, replaced by **dhcpacl_name**.

dhcpacl_data_id

The database identifier (ID) of the DHCPv4 ACL entry, a unique numeric key value automatically incremented when you add a DHCPv4 ACL entry. Use the ID to specify the DHCPv4 ACL entry of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

acl_data_id

Deprecated, replaced by **dhcpacl_data_id**.

acl_data_value

The value of the DHCPv4 ACL entry.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dhcpacl_data_name

Deprecated, replaced by **acl_data_value**.

acl_data_name

Deprecated, replaced by **acl_data_value**.

dhcpoption_name

The name of the DHCPv4 option, it must be preceded by *option* and a space as follows: *option <option-name>*. Specify an already defined DHCPv4 option to edit it.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

option

Deprecated, replaced by **dhcpoption_name**.

dhcpoption_value

The value of the DHCPv4 option.

| | | | |
|---------------|--------|----------------|------|
| Type | String | Maximum length | 4000 |
| Default value | N/A | Can be edited | Yes |

value

Deprecated, replaced by **dhcpoption_value**.

dhcpoption_type

The type of the DHCPv4 option.

| | | | |
|---------------|---|----------------|-----|
| Type | Fixed value: global scope subnet acl acl_data group range host | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

option_type

Deprecated, replaced by **dhcpoption_type**.

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 29. DHCPv6 Option

Name

dhcp6_option6_add — Add/Edit/Delete a DHCP option on DHCPv6 objects

Description

This service allows you to add, edit or delete DHCPv6 options on all DHCP objects, except leases, failover channels and shared networks. The service `dhcp6_option6_delete` does not exist.

- If no identifier is specified, a new option is created.
- If an existing identifier is specified:
 - The value specified in input edits the option.
 - The option specified without value is deleted.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** `(dhcption6_type && dhcption6_name && dhcption6_value && ((dhcp6_id || dhcp6_name) || (dhcpscope6_id || (scope6_name && (dhcp6_id || dhcp6_name || hostaddr)) || (dhcpscope6_id || (scope6_name && (dhcp6_id || dhcp6_name || hostaddr)) || (dhcpgroup6_id || (group6_name && (dhcp6_id || dhcp6_name || hostaddr)) || (dhcphost6_id || (host6_name && (dhcp6_id || dhcp6_name || hostaddr)))))))`
- **Editing:** `(dhcption6_type && dhcption6_name && ((dhcp6_id || dhcp6_name) || (dhcpscope6_id || (scope6_name && (dhcp6_id || dhcp6_name || hostaddr)) || (dhcpscope6_id || (scope6_name && (dhcp6_id || dhcp6_name || hostaddr)) || (dhcpgroup6_id || (group6_name && (dhcp6_id || dhcp6_name || hostaddr)) || (dhcphost6_id || (host6_name && (dhcp6_id || dhcp6_name || hostaddr)))))))`

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpscope6_id

The database identifier (ID) of the DHCPv6 scope, a unique numeric key value automatically incremented when you add a DHCPv6 scope. Use the ID to specify the DHCPv6 scope of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpscope6_name

The name of the DHCPv6 scope.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dhcphost6_id

The database identifier (ID) of the DHCPv6 static, a unique numeric key value automatically incremented when you add a DHCPv6 static. Use the ID to specify the DHCPv6 static of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcphost6_name

The name of the DHCPv6 static.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dhcpgroup6_id

The database identifier (ID) of the DHCPv6 group, a unique numeric key value automatically incremented when you add a DHCPv6 group. Use the ID to specify the DHCPv6 group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpgroup6_name

The name of the DHCPv6 group.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

dhcpacl6_id

The database identifier (ID) of the DHCPv6 ACL, a unique numeric key value automatically incremented when you add a DHCPv6 ACL. Use the ID to specify the DHCPv6 ACL of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpacl6_name

The name of the DHCPv6 ACL.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

dhcpacl6_data6_id

The database identifier (ID) of the DHCPv6 ACL entry, a unique numeric key value automatically incremented when you add a DHCPv6 ACL entry. Use the ID to specify the DHCPv6 ACL entry of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcption6_name

The name of the DHCPv6 option. Use the name to specify which DHCPv6 option to edit.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dhcption6_value

The value of the DHCPv6 option.

| | | | |
|---------------|--------|----------------|------|
| Type | String | Maximum length | 4000 |
| Default value | N/A | Can be edited | Yes |

dhcption6_type

The type of the DHCPv6 option.

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: global scope6 group6 acl6 acl6_data6 host6 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 30. DHCPv4 ACL and ACL Entry

Name

dhcp_acl_add — Add/Edit a DHCP ACL

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcpclass_name && (dhcp_id || dhcp_name || hostaddr))
- **Editing:** (dhcpclass_id || (dhcpclass_name && (dhcp_id || dhcp_name || hostaddr)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpclass_id

The database identifier (ID) of the DHCPv4 ACL, a unique numeric key value automatically incremented when you add a DHCPv4 ACL. Use the ID to specify the DHCPv4 ACL of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpclass_name

The name of the DHCPv4 ACL, each DHCPv4 ACL must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

dhcpclass_match

The ACL rule associated with the DHCPv4 ACL, as follows: <match if (substring(option agent.remote-id,0,6) = "dslam1");>

| | | | |
|---------------|--------|----------------|------|
| Type | String | Maximum length | 4000 |
| Default value | N/A | Can be edited | Yes |

dhcpclass_spawnwith

The spawning class associated with the DHCPv4 ACL.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dhcpclass_leaselimit

The lease limit of the DHCPv4 ACL, in seconds.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpclass_statement

The statement associated with the DHCPv4 ACL.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp_class_list — List the DHCP ACLs

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 30.1. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active (M)*, *passive (B)* or *N/A (#)*.

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpclass_id

The database identifier (ID) of the DHCPv4 ACL.

dhcpclass_name

The name of the DHCPv4 ACL.

dhcpclass_match

The ACL rule associated with the DHCPv4 ACL, as follows: <match if (substring(option agent.remote-id,0,6) = "dslam1");>

dhcpclass_spawnwith

The spawning class associated with the DHCPv4 ACL.

dhcpclass_leaselimit

The lease limit of the DHCPv4 ACL.

dhcpclass_statement

The statement associated with the DHCPv4 ACL.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

Name

dhcp_class_info — Display the properties of a DHCP ACL

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpclass_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpclass_id

The database identifier (ID) of the DHCPv4 ACL, a unique numeric key value automatically incremented when you add a DHCPv4 ACL. Use the ID to specify the DHCPv4 ACL of your choice.

Output Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 30.2. dhcp_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active* (*M*), *passive* (*B*) or *N/A* (#).

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpclass_id

The database identifier (ID) of the DHCPv4 ACL.

dhcpclass_name

The name of the DHCPv4 ACL.

dhcpclass_match

The ACL rule associated with the DHCPv4 ACL, as follows: <match if (substring(option agent.remote-id,0,6) = "dslam1");>

dhcpclass_spawnwith

The spawning class associated with the DHCPv4 ACL.

dhcpclass_leaselimit

The lease limit of the DHCPv4 ACL.

dhcpclass_statement

The statement associated with the DHCPv4 ACL.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

Name

dhcp_class_count — Count the number of DHCP ACLs

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp_acl_delete — Delete a DHCP ACL

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcpclass_id || (dhcpclass_name && (dhcp_id || dhcp_name || hostaddr)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpclass_id

The database identifier (ID) of the DHCPv4 ACL, a unique numeric key value automatically incremented when you add a DHCPv4 ACL. Use the ID to specify the DHCPv4 ACL of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

acl_id

Deprecated, replaced by **dhcpclass_id**.

dhcpclass_name

The name of the DHCPv4 ACL.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

acl_name

Deprecated, replaced by **dhcpclass_name**.

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp_acl_data_add — Add/Edit a DHCP ACL entry

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcpsubclass_value && (dhcpclass_id || (dhcpclass_name && (dhcp_id || dhcp_name || hostaddr))))
- **Editing:** ((dhcpsubclass_id || dhcpsubclass_value) && (dhcpclass_id || (dhcpclass_name && (dhcp_id || dhcp_name || hostaddr))))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpsubclass_leaselimit

The lease limit of the DHCPv4 ACL entry, in seconds.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpsubclass_id

The database identifier (ID) of the DHCPv4 ACL entry, a unique numeric key value automatically incremented when you add a DHCPv4 ACL entry. Use the ID to specify the DHCPv4 ACL entry of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpclass_id

The database identifier (ID) of the DHCPv4 ACL, a unique numeric key value automatically incremented when you add a DHCPv4 ACL. Use the ID to specify the DHCPv4 ACL of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpclass_name

The name of the DHCPv4 ACL.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp_subclass_list — List the DHCP ACL entries

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dhcpsubclass_id

The database identifier (ID) of the DHCPv4 ACL entry.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 30.3. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active* (*M*), *passive* (*B*) or *N/A* (#).

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpclass_name

The name of the DHCPv4 ACL.

dhcpclass_id

The database identifier (ID) of the DHCPv4 ACL.

dhcpsubclass_value

The value of the DHCPv4 ACL entry.

dhcpsubclass_leaselimit

The lease limit of the DHCPv4 ACL entry.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

Example

In the example below, we call the service **dhcp_subclass_list** with PHP (cURL) using the clause **ORDERBY** to sort the ACL entries based on: the DHCP server they belong to, in descending order; the name of the ACL they belong to; and finally on their value, in descending order.

Example 30.1. Calling the service dhcp_subclass_list using PHP and ORDERBY

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/dhcp_subclass_list?ORDERBY=".
                    "dhcp_name%20DESC%2C%20dhcpclass_name%2C%20dhcpsubclass_value%20DESC",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Name

dhcp_subclass_info — Display the properties of a DHCP ACL entry

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpsubclass_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpsubclass_id

The database identifier (ID) of the DHCPv4 ACL entry, a unique numeric key value automatically incremented when you add a DHCPv4 ACL entry. Use the ID to specify the DHCPv4 ACL entry of your choice.

Output Parameters

dhcpsubclass_id

The database identifier (ID) of the DHCPv4 ACL entry.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 30.4. dhcp_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |

| Type | Description |
|-------|---------------------------------------|
| vdhcp | EfficientIP DHCPv4 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active* (*M*), *passive* (*B*) or *N/A* (#).

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpclass_name

The name of the DHCPv4 ACL.

dhcpclass_id

The database identifier (ID) of the DHCPv4 ACL.

dhcpsubclass_value

The value of the DHCPv4 ACL entry.

dhcpsubclass_leaselimit

The lease limit of the DHCPv4 ACL entry.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

Name

dhcp_subclass_count — Count the number of DHCP ACL entries

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp_acl_data_delete — Delete a DHCP ACL entry

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcpsubclass_id || (dhcpsubclass_value && (dhcpclass_id || (dhcpclass_name && (dhcp_id || dhcp_name || hostaddr)))))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_addr

Deprecated, replaced by **hostaddr**.

dhcpclass_id

The database identifier (ID) of the DHCPv4 ACL, a unique numeric key value automatically incremented when you add a DHCPv4 ACL. Use the ID to specify the DHCPv4 ACL of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

acl_id

Deprecated, replaced by **dhcpclass_id**.

dhcpclass_name

The name of the DHCPv4 ACL.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

acl_name

Deprecated, replaced by **dhcpclass_name**.

dhcpsubclass_id

The database identifier (ID) of the DHCPv4 ACL entry, a unique numeric key value automatically incremented when you add a DHCPv4 ACL entry. Use the ID to specify the DHCPv4 ACL entry of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

acl_data_id

Deprecated, replaced by **dhcpsubclass_id**.

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 31. DHCPv6 ACL and ACL Entry

Name

dhcp6_acl6_add — Add/Edit a DHCPv6 ACL

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcpclass6_name && (dhcp6_id || dhcp6_name || hostaddr))
- **Editing:** ((dhcpclass6_id || dhcpclass6_name) && (dhcp6_id || dhcp6_name || hostaddr))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpclass6_id

The database identifier (ID) of the DHCPv6 ACL, a unique numeric key value automatically incremented when you add a DHCPv6 ACL. Use the ID to specify the DHCPv6 ACL of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpclass6_name

The name of the DHCPv6 ACL, each DHCPv6 ACL must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

dhcpclass6_match

The ACL rule associated with the DHCPv6 ACL, as follows: <match if (substring(option agent.remote-id,0,6) = "dslam1");>

| | | | |
|---------------|--------|----------------|------|
| Type | String | Maximum length | 4000 |
| Default value | N/A | Can be edited | Yes |

dhcpclass6_spawnwith

The spawning class associated with the DHCPv6 ACL.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dhcpclass6_statement

The statement associated with the DHCPv6 ACL.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp6_class6_list — List the DHCPv6 ACLs

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter **offset** must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter **limit** must be specified in **lowercase**.

Output Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 31.1. *dhcp6_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active* (*M*), *passive* (*B*) or *N/A* (#).

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. *0* indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpclass6_id

The database identifier (ID) of the DHCPv6 ACL.

dhcpclass6_name

The name of the DHCPv6 ACL.

dhcpclass6_match

The ACL rule associated with the DHCPv6 ACL, as follows: <match if (substring(option agent.remote-id,0,6) = "dslam1");>

dhcpclass6_spawnwith

The spawning class associated with the DHCPv6 ACL.

dhcpclass6_statement

The statement associated with the DHCPv6 ACL.

delayed_time

The delay of creation/deletion status. *1* indicates that the object is not created/deleted yet.

Name

dhcp6_class6_info — Display the properties of a DHCPv6 ACL

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpclass6_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpclass6_id

The database identifier (ID) of the DHCPv6 ACL, a unique numeric key value automatically incremented when you add a DHCPv6 ACL. Use the ID to specify the DHCPv6 ACL of your choice.

Output Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 31.2. dhcp6_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active* (*M*), *passive* (*B*) or *N/A* (#).

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpclass6_id

The database identifier (ID) of the DHCPv6 ACL.

dhcpclass6_name

The name of the DHCPv6 ACL.

dhcpclass6_match

The ACL rule associated with the DHCPv6 ACL, as follows: <match if (substring(option agent.remote-id,0,6) = "dslam1");>

dhcpclass6_spawnwith

The spawning class associated with the DHCPv6 ACL.

dhcpclass6_statement

The statement associated with the DHCPv6 ACL.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

Name

dhcp6_class6_count — Count the number of DHCPv6 ACLs

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp6_acl6_delete — Delete a DHCPv6 ACL

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcpclass6_id || (dhcpclass6_name && (dhcp6_id || dhcp6_name || hostaddr)))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpclass6_id

The database identifier (ID) of the DHCPv6 ACL, a unique numeric key value automatically incremented when you add a DHCPv6 ACL. Use the ID to specify the DHCPv6 ACL of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpclass6_name

The name of the DHCPv6 ACL.

| Type | String | Maximum length | 64 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp6_acl_data6_add — Add/Edit a DHCPv6 ACL entry

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dhcpsubclass6_value && (dhcpclass6_id || (dhcpclass6_name && (dhcp6_id || dhcp6_name || hostaddr))))
- **Editing:** ((dhcpsubclass6_id || dhcpsubclass6_value) && (dhcpclass6_id || (dhcpclass6_name && (dhcp6_id || dhcp6_name || hostaddr))))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpsubclass6_value

The value of the DHCPv6 ACL entry.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpsubclass6_id

The database identifier (ID) of the DHCPv6 ACL entry, a unique numeric key value automatically incremented when you add a DHCPv6 ACL entry. Use the ID to specify the DHCPv6 ACL entry of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpclass6_id

The database identifier (ID) of the DHCPv6 ACL, a unique numeric key value automatically incremented when you add a DHCPv6 ACL. Use the ID to specify the DHCPv6 ACL of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpclass6_name

The name of the DHCPv6 ACL.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dhcp6_subclass6_list — List the DHCPv6 ACL entries

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter **offset** must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter **limit** must be specified in **lowercase**.

Output Parameters

dhcpsubclass6_id

The database identifier (ID) of the DHCPv6 ACL entry.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 31.3. *dhcp6_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active* (*M*), *passive* (*B*) or *N/A* (#).

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpclass6_name

The name of the DHCPv6 ACL.

dhcpclass6_id

The database identifier (ID) of the DHCPv6 ACL.

dhcpsubclass6_value

The value of the DHCPv6 ACL entry.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

Example

In the example below, we call the service **dhcp_subclass_list** with PHP (cURL) using the clause **ORDERBY** to sort the ACL entries based on: the DHCP server they belong to, in descending order; the name of the ACL they belong to; and finally on their value, in descending order.

Example 31.1. Calling the service dhcp_subclass_list using PHP and ORDERBY

```
<?php
```

```
$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/dhcp_subclass_list?ORDERBY=".
                    "dhcp_name%20DESC%2C%20dhcpclass_name%2C%20dhcpsubclass_value%20DESC",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Name

dhcp6_subclass6_info — Display the properties of a DHCPv6 ACL entry

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpsubclass6_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpsubclass6_id

The database identifier (ID) of the DHCPv6 ACL entry, a unique numeric key value automatically incremented when you add a DHCPv6 ACL entry. Use the ID to specify the DHCPv6 ACL entry of your choice.

Output Parameters

dhcpsubclass6_id

The database identifier (ID) of the DHCPv6 ACL entry.

dhcp6_id

The database identifier (ID) of the DHCPv6 server the object belongs to.

dhcp6_name

The name of the DHCPv6 server the object belongs to.

dhcp6_type

The type of the DHCPv6 server the object belongs to:

Table 31.4. dhcp6_type possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| vdhcp | EfficientIP DHCPv6 smart architecture |

cluster_role

The role of the server the object belongs to in the cluster, either *active* (*M*), *passive* (*B*) or *N/A* (#).

vdhcp6_parent_id

The database identifier (ID) of the DHCPv6 smart architecture managing the DHCPv6 server the object belongs to. *0* indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcpclass6_name

The name of the DHCPv6 ACL.

dhcpclass6_id

The database identifier (ID) of the DHCPv6 ACL.

dhcpsubclass6_value

The value of the DHCPv6 ACL entry.

delayed_time

The delay of creation/deletion status. *1* indicates that the object is not created/deleted yet.

Name

dhcp6_subclass6_count — Count the number of DHCPv6 ACL entries

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp6_acl_data6_delete — Delete a DHCPv6 ACL entry

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcpsubclass6_id || (dhcpsubclass6_value && (dhcpclass6_id || (dhcpclass6_name && (dhcp6_id || dhcp6_name || hostaddr)))))

Input Parameters

dhcp6_id

The database identifier (ID) of the DHCPv6 server, a unique numeric key value automatically incremented when you add a DHCPv6 server. Use the ID to specify the DHCPv6 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp6_name

The name of the DHCPv6 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DHCP server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpclass6_id

The database identifier (ID) of the DHCPv6 ACL, a unique numeric key value automatically incremented when you add a DHCPv6 ACL. Use the ID to specify the DHCPv6 ACL of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpclass6_name

The name of the DHCPv6 ACL.

| Type | String | Maximum length | 64 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpsubclass6_id

The database identifier (ID) of the DHCPv6 ACL entry, a unique numeric key value automatically incremented when you add a DHCPv6 ACL entry. Use the ID to specify the DHCPv6 ACL entry of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dhcpsubclass6_value

The value of the DHCPv6 ACL entry.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 32. DHCPv4 Failover Channel

Name

dhcp_failover_list — List the DHCPv4 failover channels

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter **offset** must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter **limit** must be specified in **lowercase**.

Output Parameters

dhcpfailover_id

The database identifier (ID) of the DHCPv4 failover channel, a unique numeric key value automatically incremented when you add a failover channel.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 32.1. *dhcp_type* possible values

| Type | Description |
|-------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcp_state

The status of the DHCPv4 smart architecture.

cluster_role

The role of the server the object belongs to in the cluster, either *active* (M), *passive* (B) or *N/A* (#).

dhcpfailover_name

The name of the DHCPv4 failover channel.

dhcpfailover_addr

The IP address of the primary DHCPv4 server.

dhcpfailover_port

The port number of the primary DHCPv4 server.

peer_dhcp_id

The database identifier (ID) of the secondary DHCPv4 server.

dhcpfailover_peer_addr

The IP address of the secondary DHCPv4 server.

dhcpfailover_peer_port

The port number of the secondary DHCPv4 server.

dhcpfailover_split

Internal use. Not documented.

dhcpfailover_state

The status of the DHCPv4 failover channel, either *startup*, *normal*, *communications-interrupted* or *recover-wait*.

dhcpfailover_type

The type of the DHCPv4 failover channel, either *primary* or *secondary*.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

ip_addr

The Management IP address of the DHCPv4 server, the IPv4 address configured when adding the server, on which the failover channel is configured, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server, the IPv6 address configured when adding the server, on which the failover channel is configured, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

dhcpfailover_auto_partner_down

The time after which the DHCPv4 failover channel automatically switches to *partner-down* after being in *communication-interrupted* state, in hours.

dhcpfailover_mclt

The maximum client lead time (MCLT) of the failover channel, in seconds. It indicates for how long each DHCP server can extend the lease of a client, beyond the time known by its partner server.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 32.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

Name

dhcp_failover_info — Display the properties of a DHCPv4 failover channel

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dhcpfailover_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dhcpfailover_id

The database identifier (ID) of the DHCPv4 failover channel, a unique numeric key value automatically incremented when you add a DHCPv4 failover channel. Use the ID to specify the DHCPv4 failover channel of your choice.

Output Parameters

dhcpfailover_id

The database identifier (ID) of the DHCPv4 failover channel, a unique numeric key value automatically incremented when you add a failover channel.

dhcp_name

The name of the DHCPv4 server the object belongs to.

dhcp_id

The database identifier (ID) of the DHCPv4 server the object belongs to.

dhcp_type

The type of the DHCPv4 server the object belongs to:

Table 32.3. `dhcp_type` possible values

| Type | Description |
|------|---|
| ipm | EfficientIP or EfficientIP Package server |

| Type | Description |
|-------|---------------------------------------|
| msrpc | Microsoft Windows DHCP server |
| vdhcp | EfficientIP DHCPv4 smart architecture |

vdhcp_parent_id

The database identifier (ID) of the DHCPv4 smart architecture managing the DHCPv4 server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dhcp_state

The status of the DHCPv4 smart architecture.

cluster_role

The role of the server the object belongs to in the cluster, either *active* (M), *passive* (B) or N/A (#).

dhcpfailover_name

The name of the DHCPv4 failover channel.

dhcpfailover_addr

The IP address of the primary DHCPv4 server.

dhcpfailover_port

The port number of the primary DHCPv4 server.

peer_dhcp_id

The database identifier (ID) of the secondary DHCPv4 server.

dhcpfailover_peer_addr

The IP address of the secondary DHCPv4 server.

dhcpfailover_peer_port

The port number of the secondary DHCPv4 server.

dhcpfailover_split

Internal use. Not documented.

dhcpfailover_state

The status of the DHCPv4 failover channel, either *startup*, *normal*, *communications-interrupted* or *recover-wait*.

dhcpfailover_type

The type of the DHCPv4 failover channel, either *primary* or *secondary*.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

ip_addr

The Management IP address of the DHCPv4 server, the IPv4 address configured when adding the server, on which the failover channel is configured, in hexadecimal format.

ip6_addr

The Management IP address of the DHCPv4 server, the IPv6 address configured when adding the server, on which the failover channel is configured, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

dhcpfailover_auto_partner_down

The time after which the DHCPv4 failover channel automatically switches to *partner-down* after being in *communication-interrupted* state, in hours.

dhcpfailover_mclt

The maximum client lead time (MCLT) of the failover channel, in seconds. It indicates for how long each DHCP server can extend the lease of a client, beyond the time known by its partner server.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 32.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

Name

dhcp_failover_count — Count the number of DHCPv4 failover channels

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dhcp_failover_set_partner_down — Set a DHCPv4 failover channel to PARTNER-DOWN

Description

This service allows you to set a DHCPv4 failover channel to PARTNER-DOWN.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dhcpfailover_id || (dhcpfailover_name && (dhcp_id || dhcp_name)))

Input Parameters

dhcp_id

The database identifier (ID) of the DHCPv4 server, a unique numeric key value automatically incremented when you add a DHCPv4 server. Use the ID to specify the DHCPv4 server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcp_name

The name of the DHCPv4 server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpfailover_id

The database identifier (ID) of the DHCPv4 failover channel, a unique numeric key value automatically incremented when you add a DHCPv4 failover channel. Use the ID to specify the DHCPv4 failover channel of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dhcpfailover_name

The name of the DHCPv4 failover channel.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Part IV. DNS Services

Table of Contents

| | |
|-------------------------------|-----|
| 33. DNS Server | 578 |
| dns_server_list | 579 |
| dns_server_info | 587 |
| dns_server_count | 595 |
| dns_smart_member_add | 596 |
| dns_smart_member_delete | 598 |
| 34. DNS View | 600 |
| dns_view_add | 601 |
| dns_view_list | 606 |
| dns_view_info | 610 |
| dns_view_count | 614 |
| group_dnsview_add | 615 |
| group_dnsview_delete | 617 |
| dns_view_delete | 619 |
| dns_view_param_add | 621 |
| dns_view_param_list | 623 |
| dns_view_param_info | 625 |
| dns_view_param_count | 627 |
| dns_view_param_delete | 628 |
| 35. DNS Zone | 630 |
| dns_zone_add | 631 |
| dns_zone_list | 639 |
| dns_zone_info | 647 |
| dns_zone_count | 654 |
| dns_zone_groupby | 655 |
| dns_zone_groupby_count | 657 |
| group_dnszone_add | 659 |
| group_dnszone_delete | 662 |
| dns_zone_delete | 665 |
| dns_zone_param_add | 667 |
| dns_zone_param_list | 669 |
| dns_zone_param_info | 671 |
| dns_zone_param_count | 673 |
| dns_zone_param_delete | 674 |
| 36. DNS Resource Record | 676 |
| dns_rr_add | 677 |
| dns_rr_list | 686 |
| dns_rr_info | 695 |
| dns_rr_count | 703 |
| dns_rr_groupby | 704 |
| dns_rr_groupby_count | 706 |
| dns_rr_delete | 708 |
| 37. DNS ACL | 713 |
| dns_acl_add | 714 |
| dns_acl_list | 717 |
| dns_acl_info | 719 |
| dns_acl_count | 720 |
| dns_acl_delete | 721 |
| 38. TSIG Key | 723 |
| dns_key_add | 724 |
| dns_key_list | 727 |

| | |
|-------------------------------|-----|
| dns_key_info | 729 |
| dns_key_count | 731 |
| dns_key_delete | 732 |
| 39. DNSSEC | 734 |
| dnssec_zone_keys_list | 735 |
| dnssec_zone_keys_info | 738 |
| dnssec_enable_sign_zone | 740 |

Chapter 33. DNS Server

Name

dns_server_list — List the DNS servers

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dns_role

The role of the DNS server in the smart architecture, either *master*, *hidden-master*, *pseudo-master* or *slave*.

modif_count

Internal use. Not documented.

dns_cloud_private

The type of the DNS Amazon Route 53 or Azure server, either Public (0) or Private (1). If the server is not a cloud server, 0 is returned.

aws_keyid

The AWS access key identifier (ID) of the DNS server.

aws_use_role

The configuration of the option *Use a role* of the Amazon Route 53 DNS server, either enabled (1) or not (0). If enabled, the role can rely on ARN, an external ID or a session.

aws_role_arn

The Amazon Resource Name (ARN) of the role used for the Amazon Route 53 DNS server.

aws_role_external_id

The external ID of the role used for the Amazon Route 53 DNS server.

aws_role_session_name

The session name of the role used for the Amazon Route 53 DNS server.

aws_delegation_set

The reusable delegation set ID configured on the Amazon Route 53 public server.

az_tenantid

For Microsoft Azure servers, the tenant ID of the DNS server.

az_keyid

For Microsoft Azure servers, the Azure Application ID of the DNS server.

az_subscriptionid

For Microsoft Azure servers, the subscription ID of the DNS server.

az_group

For Microsoft Azure servers, the resource group of the DNS server.

connectionprofile_name

The name of the connection profile used as connection method for the DNS server.

ipmdns_is_package

The DNS server package information. Y for an EfficientIP Package server, N for an appliance or virtual machine, U the package information is irrelevant. For servers with a **dns_type** set to *ipm*, U indicates either EfficientIP Packages or appliances/virtual machines.

ipmdns_https_login

Internal use. Not documented.

ipmdns_protocol

Internal use. Not documented.

ipmdns_type

The engine type of the DNS server: *named* (BIND engine), *nsd* (NSD engine) or *unbound* (Unbound engine).

tree_path

The database path toward the server as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

isolated

A way to determine if the server can update any other module (1).

windns_port

Internal use. Not documented.

windns_use_ssl

Internal use. Not documented.

windns_protocol

Internal use. Not documented.

dns_notify

The notify status of the DNS server:

Table 33.1. *dns_notify* possible values

| Status | Description |
|----------|---|
| no | No notify message is sent when changes are performed in the master zones. |
| yes | The notify messages are sent to the target of the NS records of the master zone. They are also sent to the IP address(es) specified in the parameter dns_also_notify . |
| explicit | The notify messages are only sent to the IP address(es) specified in the parameter dns_also_notify . |

dns_also_notify

The IP address and port of the DNS server managing the smart architecture. If the parameter **dns_notify** is set to *yes* or *explicit*, the server specified is instantly notified of any slave zones updates.

dns_allow_query_cache

The ACL values associated with the allow-query-cache configuration of the DNS server, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dns_allow_query

The ACL values associated with the allow-query configuration of the DNS server, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dns_allow_transfer

The ACL values associated with the allow-transfer configuration of the DNS server, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dns_allow_recursion

The ACL values associated with the allow-recursion configuration of the DNS server, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dns_recursion

The recursion status of the DNS server:

Table 33.2. dns_recursion possible values

| Status | Description |
|--------|--|
| no | The server only provides iterative query behavior - normally resulting in a referral. If the answer to the query already exists in the cache it will be returned whatever the value of this statement. |
| yes | The server always provides recursive query behavior if requested by the client. |

dns_forwarders

The IP address(es) of the forwarder(s) associated with the DNS server. It lists the DNS servers to which any unknown zone should be sent, as follows: <ip_address1>;<ip_address2>;... .

dns_forward

The forwarding mode of the DNS server. No value indicates that the forwarding is disabled:

Table 33.3. dns_forward possible values

| Status | Description |
|--------|--|
| first | The server sends the queries to the forwarder(s). If no answer is returned, it attempts to answer the queries on its own. |
| only | The server only forwards the queries to the forwarder(s). Required by some reverse forward zones (e.g., in the case of private addresses). |

snmp_id

Internal use. Not documented.

ldap_user

For Microsoft Windows servers, the login of the user communicating with the DNS server.

ldap_domain

For Microsoft Windows servers, the domain of the DNS server.

vdns_public_ns_list

The list of the published name servers associated with the DNS smart architecture, as follows: <ns1>;<ns2>;... .

tree_level

The database level of the server. 0 indicates the server is managed on its own, 1 indicates it is managed by a smart architecture.

total_vdns_members

The total number of servers managed by the DNS smart architecture.

vdns_members_name

The list of the servers managed by the DNS smart architecture, as follows: <dns_name>;<dns_name>;... .

vdns_arch

The type of the DNS smart architecture:

Table 33.4. vdns_arch possible values

| Status | Description |
|-------------|--------------|
| masterslave | Master/Slave |
| stealth | Stealth |
| multimaster | Multi-Master |

| Status | Description |
|--------|---------------|
| single | Single-Server |
| farm | Farm |

vdns_parent_name

The name of the DNS smart architecture managing the DNS server. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdns_parent_arch

The type of the DNS smart architecture managing the DNS server. No value indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdns_parent_id

The database identifier (ID) of the DNS smart architecture managing the DNS server. 0 indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server.

dns_state

The status of the DNS server:

Table 33.5. dns_state possible values

| Status | Description |
|--------|---|
| ER | The license used in SOLIDserver is not compliant with the added server: the license is invalid. |
| ES | The server configuration could not be parsed properly. |
| ET | The server does not answer anymore due to a scheduled configuration of the server. |
| IC | The SSL credentials are invalid |
| IP | The account used to add the Microsoft Windows DNS server does not have sufficient privileges to manage it. |
| IR | SOLIDserver cannot resolve the AWS DNS service. The Amazon services are unreachable and the Amazon Route 53 server cannot be managed. Make sure that the DNS resolvers declared on the page <i>Network configuration</i> are valid. |
| IS | There was a setting error during the server declaration. For instance, some settings were added to a server that does not support them or a smart architecture is not managing any physical server. |
| IT | The server editing performed from the GUI is not pushed to the server because SOLIDserver time and date are incorrect. You must use the UTC system on the appliance, especially when managing Amazon Route 53 servers. |
| LS | The server configuration is not viable. |
| N | The server does not have a status as it has not synchronized yet. |
| UE | An error occurred that SOLIDserver could not identify. |
| Y | The server is operational. |

querylog_state

The DNS querylog status. 1 indicates that the DNS server querylog is enabled.

dns_synching

The synchronization status of the DNS server. 1 indicates that the server is currently being synchronized.

dns_name

The name of the DNS server.

dns_comment

The description of the DNS server.

dns_type

The type of the DNS server:

Table 33.6. dns_type possible values

| Type | Description |
|----------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msdaemon | Microsoft Windows DNS server |
| aws | Amazon Route 53 server |
| other | Generic DNS server |
| vdns | EfficientIP DNS smart architecture |

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

ip_addr

The IPv4 address of the DNS server, in hexadecimal format.

ip6_addr

The IPv6 address of the DNS server, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

dns_class_name

The name of the class applied to the DNS server, it can be preceded by the class directory.

dns_version

The version details of the DNS server.

dns_key_name

The name of the DNS TSIG key associated with the DNS server.

dns_key_value

The value of the TSIG key associated with the DNS server.

dns_key_proto

The encryption protocol of the TSIG key associated with the DNS server.

dns_hybrid

Internal use. Not documented.

dns_force_hybrid

Internal use. Not documented.

dns_cloud

Internal use. Not documented.

dns_rpz_recursive_only

The configuration of the option *Enable recursive-only on the server*. It applies to the RPZ zones of the server, if it is not set at zone level. By default it is set to 1 (yes), the server only processes policies on recursive queries.

dns_rpz_break_dnssec

The configuration of the option *Enable break-dnssec on the server*. It applies to the RPZ zones of the server, if it is not set at zone level. By default it is set to 0 (no), if set to 1 (yes) the server processes policies on all DNSSEC queries.

dns_rpz_qname_wait_recurse

The configuration of the option *Enable qname-wait-recurse on the server*. It applies to the RPZ zones of the server, if it is not set at zone level. By default it is set to 0 (no), if set to 1 (yes) the server only process policies when the results of any query are available.

dns_rpz_max_policy_ttl

The configuration of the option *Server max policy TTL*, i.e. the number of seconds of the max policy Time To Live of the server. It applies to the RPZ zones of the server, if it is not set at zone level. By default it is empty, i.e. set to 5 seconds.

dns_rpz_min_ns_dots

The configuration of the option *Server minimum dot separators of the server*, i.e. the minimum number of dot separators required in any QNAME to process policies. It applies to the RPZ zones of the server, if it is not set at zone level. By default it is empty, i.e. set to 1.

gss_keytab_id

The database identifier (ID) of the DNS GSS-TSIG keytab.

gss_enabled

The GSS-TSIG status of the DNS server. 1 indicates that GSS-TSIG is enabled on the server.

reverse_proxy_conf

The URL of the HTTP(S) reverse proxy server that forwards client queries to the DNS server, if you configured one.

stat_enabled

Internal use. Not documented.

stat_period

Internal use. Not documented.

stat_niceness

Internal use. Not documented.

stat_time

Internal use. Not documented.

dnsblast_enabled

The status of the service DNS Guardian, either enabled (1) or disabled (0).

dnsblast_status

The status of the Guardian server, either OK (1), Stopped (2), Invalid Credentials (4) or Timeout (5).

dnssec_validation

The DNSSEC resolution status of the DNS server. yes indicates it is enabled.

dns gslb_supported

The license GSLB activation status. 1 indicates your license includes GSLB and your appliance supports it.

dns guardian_supported

The license Guardian activation status. 1 indicates your license includes Guardian and your appliance supports its latest features.

dns guardian gui management supported

Internal use. Not documented.

guardian_stats_only_supported

A way to determine if the server only retrieves Guardian statistics (1) or not (0). On appliances where the service *DNS Guardian / GSLB server* is not configured, the server cannot be configured with Guardian options, it can only retrieve statistics.

dns_vpc_list

The list of cloud networks configured on the DNS private server, separated by a comma:

- For an Azure server, it returns the list of virtual networks.
- For an Amazon Route 53 server, it returns the list of Virtual Private Cloud (VPC).

dnsblast_push_status

Internal use. Not documented.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 33.7. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dns_class_parameters

The class parameters applied to the DNS server and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dns_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dns_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dns_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

dns_server_info — Display the properties of a DNS server

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dns_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

Output Parameters

dns_role

The role of the DNS server in the smart architecture, either `master`, `hidden-master`, `pseudo-master` or `slave`.

modif_count

Internal use. Not documented.

dns_cloud_private

The type of the DNS Amazon Route 53 or Azure server, either Public (0) or Private (1). If the server is not a cloud server, 0 is returned.

aws_keyid

The AWS access key identifier (ID) of the DNS server.

aws_use_role

The configuration of the option `Use a role` of the Amazon Route 53 DNS server, either enabled (1) or not (0). If enabled, the role can rely on ARN, an external ID or a session.

aws_role_arn

The Amazon Resource Name (ARN) of the role used for the Amazon Route 53 DNS server.

aws_role_external_id

The external ID of the role used for the Amazon Route 53 DNS server.

aws_role_session_name

The session name of the role used for the Amazon Route 53 DNS server.

aws_delegation_set

The reusable delegation set ID configured on the Amazon Route 53 public server.

az_tenantid

For Microsoft Azure servers, the tenant ID of the DNS server.

az_keyid

For Microsoft Azure servers, the Azure Application ID of the DNS server.

az_subscriptionid

For Microsoft Azure servers, the subscription ID of the DNS server.

az_group

For Microsoft Azure servers, the resource group of the DNS server.

connectionprofile_name

The name of the connection profile used as connection method for the DNS server.

ipmdns_is_package

The DNS server package information. *Y* for an EfficientIP Package server, *N* for an appliance or virtual machine, *U* the package information is irrelevant. For servers with a **dns_type** set to *jpm*, *U* indicates either EfficientIP Packages or appliances/virtual machines.

ipmdns_https_login

Internal use. Not documented.

ipmdns_protocol

Internal use. Not documented.

ipmdns_type

The engine type of the DNS server: *named* (BIND engine), *nsd* (NSD engine) or *unbound* (Unbound engine).

tree_path

The database path toward the server as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

isolated

A way to determine if the server can update any other module (1).

windns_port

Internal use. Not documented.

windns_use_ssl

Internal use. Not documented.

windns_protocol

Internal use. Not documented.

dns_notify

The notify status of the DNS server:

Table 33.8. dns_notify possible values

| Status | Description |
|----------|---|
| no | No notify message is sent when changes are performed in the master zones. |
| yes | The notify messages are sent to the target of the NS records of the master zone. They are also sent to the IP address(es) specified in the parameter dns_also_notify . |
| explicit | The notify messages are only sent to the IP address(es) specified in the parameter dns_also_notify . |

dns_also_notify

The IP address and port of the DNS server managing the smart architecture. If the parameter **dns_notify** is set to *yes* or *explicit*, the server specified is instantly notified of any slave zones updates.

dns_allow_query_cache

The ACL values associated with the allow-query-cache configuration of the DNS server, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dns_allow_query

The ACL values associated with the allow-query configuration of the DNS server, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dns_allow_transfer

The ACL values associated with the allow-transfer configuration of the DNS server, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dns_allow_recursion

The ACL values associated with the allow-recursion configuration of the DNS server, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dns_recursion

The recursion status of the DNS server:

Table 33.9. dns_recursion possible values

| Status | Description |
|--------|--|
| no | The server only provides iterative query behavior - normally resulting in a referral. If the answer to the query already exists in the cache it will be returned whatever the value of this statement. |
| yes | The server always provides recursive query behavior if requested by the client. |

dns_forwarders

The IP address(es) of the forwarder(s) associated with the DNS server. It lists the DNS servers to which any unknown zone should be sent, as follows: <ip_address1>;<ip_address2>;... .

dns_forward

The forwarding mode of the DNS server. No value indicates that the forwarding is disabled:

Table 33.10. dns_forward possible values

| Status | Description |
|--------|---|
| first | The server sends the queries to the forwarder(s). If no answer is returned, it attempts to answer the queries on its own. |

| Status | Description |
|--------|--|
| only | The server only forwards the queries to the forwarder(s). Required by some reverse forward zones (e.g., in the case of private addresses). |

snmp_id

Internal use. Not documented.

ldap_user

For Microsoft Windows servers, the login of the user communicating with the DNS server.

ldap_domain

For Microsoft Windows servers, the domain of the DNS server.

vdns_public_ns_list

The list of the published name servers associated with the DNS smart architecture, as follows:
`<ns1>;<ns2>;...`

tree_level

The database level of the server. *0* indicates the server is managed on its own, *1* indicates it is managed by a smart architecture.

total_vdns_members

The total number of servers managed by the DNS smart architecture.

vdns_members_name

The list of the servers managed by the DNS smart architecture, as follows:
`<dns_name>;<dns_name>;...`

vdns_arch

The type of the DNS smart architecture:

Table 33.11. vdns_arch possible values

| Status | Description |
|-------------|---------------|
| masterslave | Master/Slave |
| stealth | Stealth |
| multimaster | Multi-Master |
| single | Single-Server |
| farm | Farm |

vdns_parent_name

The name of the DNS smart architecture managing the DNS server. # indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdns_parent_arch

The type of the DNS smart architecture managing the DNS server. No value indicates that the server is not managed by a smart architecture or is a smart architecture itself.

vdns_parent_id

The database identifier (ID) of the DNS smart architecture managing the DNS server. *0* indicates that the server is not managed by a smart architecture or is a smart architecture itself.

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server.

dns_state

The status of the DNS server:

Table 33.12. dns_state possible values

| Status | Description |
|--------|---|
| ER | The license used in SOLIDserver is not compliant with the added server: the license is invalid. |
| ES | The server configuration could not be parsed properly. |
| ET | The server does not answer anymore due to a scheduled configuration of the server. |
| IC | The SSL credentials are invalid |
| IP | The account used to add the Microsoft Windows DNS server does not have sufficient privileges to manage it. |
| IR | SOLIDserver cannot resolve the AWS DNS service. The Amazon services are unreachable and the Amazon Route 53 server cannot be managed. Make sure that the DNS resolvers declared on the page <i>Network configuration</i> are valid. |
| IS | There was a setting error during the server declaration. For instance, some settings were added to a server that does not support them or a smart architecture is not managing any physical server. |
| IT | The server editing performed from the GUI is not pushed to the server because SOLIDserver time and date are incorrect. You must use the UTC system on the appliance, especially when managing Amazon Route 53 servers. |
| LS | The server configuration is not viable. |
| N | The server does not have a status as it has not synchronized yet. |
| UE | An error occurred that SOLIDserver could not identify. |
| Y | The server is operational. |

querylog_state

The DNS querylog status. 1 indicates that the DNS server querylog is enabled.

dns_synching

The synchronization status of the DNS server. 1 indicates that the server is currently being synchronized.

dns_name

The name of the DNS server.

dns_comment

The description of the DNS server.

dns_type

The type of the DNS server:

Table 33.13. dns_type possible values

| Type | Description |
|----------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msdaemon | Microsoft Windows DNS server |
| aws | Amazon Route 53 server |
| other | Generic DNS server |
| vdns | EfficientIP DNS smart architecture |

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.

- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, `row_enabled` is set to 1 when an object is created.

ip_addr

The IPv4 address of the DNS server, in hexadecimal format.

ip6_addr

The IPv6 address of the DNS server, in hexadecimal format.

hostaddr

The human readable version of the parameter `ip_addr` or `ip6_addr`.

dns_class_name

The name of the class applied to the DNS server, it can be preceded by the class directory.

dns_version

The version details of the DNS server.

dns_key_name

The name of the DNS TSIG key associated with the DNS server.

dns_key_value

The value of the TSIG key associated with the DNS server.

dns_key_proto

The encryption protocol of the TSIG key associated with the DNS server.

dns_hybrid

Internal use. Not documented.

dns_force_hybrid

Internal use. Not documented.

dns_cloud

Internal use. Not documented.

dns_rpz_recursive_only

The configuration of the option *Enable recursive-only on the server*. It applies to the RPZ zones of the server, if it is not set at zone level. By default it is set to 1 (yes), the server only processes policies on recursive queries.

dns_rpz_break_dnssec

The configuration of the option *Enable break-dnssec on the server*. It applies to the RPZ zones of the server, if it is not set at zone level. By default it is set to 0 (no), if set to 1 (yes) the server processes policies on all DNSSEC queries.

dns_rpz_qname_wait_recurse

The configuration of the option *Enable qname-wait-recurse on the server*. It applies to the RPZ zones of the server, if it is not set at zone level. By default it is set to 0 (no), if set to 1 (yes) the server only process policies when the results of any query are available.

dns_rpz_max_policy_ttl

The configuration of the option *Server max policy TTL*, i.e. the number of seconds of the max policy Time To Live of the server. It applies to the RPZ zones of the server, if it is not set at zone level. By default it is empty, i.e. set to 5 seconds.

dns_rpz_min_ns_dots

The configuration of the option *Server minimum dot separators of the server*, i.e. the minimum number of dot separators required in any QNAME to process policies. It applies to the RPZ zones of the server, if it is not set at zone level. By default it is empty, i.e. set to 1.

gss_keytab_id

The database identifier (ID) of the DNS GSS-TSIG keytab.

gss_enabled

The GSS-TSIG status of the DNS server. 1 indicates that GSS-TSIG is enabled on the server.

reverse_proxy_conf

The URL of the HTTP(S) reverse proxy server that forwards client queries to the DNS server, if you configured one.

stat_enabled

Internal use. Not documented.

stat_period

Internal use. Not documented.

stat_niceness

Internal use. Not documented.

stat_time

Internal use. Not documented.

dnsblast_enabled

The status of the service DNS Guardian, either enabled (1) or disabled (0).

dnsblast_status

The status of the Guardian server, either OK (1), Stopped (2), Invalid Credentials (4) or Timeout (5).

dnssec_validation

The DNSSEC resolution status of the DNS server. yes indicates it is enabled.

dns gslb_supported

The license GSLB activation status. 1 indicates your license includes GSLB and your appliance supports it.

dns guardian_supported

The license Guardian activation status. 1 indicates your license includes Guardian and your appliance supports its latest features.

dns guardian_gui_management_supported

Internal use. Not documented.

guardian_stats_only_supported

A way to determine if the server only retrieves Guardian statistics (1) or not (0). On appliances where the service *DNS Guardian / GSLB* server is not configured, the server cannot be configured with Guardian options, it can only retrieve statistics.

dns_vpc_list

The list of cloud networks configured on the DNS private server, separated by a comma:

- For an Azure server, it returns the list of virtual networks.
- For an Amazon Route 53 server, it returns the list of Virtual Private Cloud (VPC).

dnsblast_push_status

Internal use. Not documented.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 33.14. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dns_class_parameters

The class parameters applied to the DNS server and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dns_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dns_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dns_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

dns_server_count — Count the number of DNS servers

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dns_smart_member_add — Add a DNS server to a smart architecture

Description

This service allows you to manage a DNS server from a smart architecture. A call can only add one existing DNS server to an existing DNS smart architecture.

Once added to a smart architecture, a server configuration must be managed from the architecture.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(vdns_id || vdns_name) && ((dns_id || dns_name || hostaddr) && dns_role)

Input Parameters

vdns_id

The database identifier (ID) of the DNS smart architecture, a unique numeric key value automatically incremented when you add a DNS smart architecture. Use the ID to specify in which smart architecture you add the DNS server.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vdns_name

The name of the DNS smart architecture in which you add the DNS server.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| Type | IPv4 address | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_role

The role of the DNS server within the smart architecture. Note that EfficientIP, EfficientIP package and Microsoft servers can assume any role; BIND/Unbound hybrid servers can only be defined as master or slave; Generic servers can only be defined as master; Amazon Route 53 and Azure servers can only be defined as master or pseudo-master.

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: master slave hidden-master pseudo-master | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_role_id

The database identifier (ID) of the **dns_role** of the DNS server within the smart architecture.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

dns_smart_member_delete — Remove a DNS server from a smart architecture

Description

This service allows you to stop managing a DNS server from a smart architecture. A call can only remove one DNS server from a DNS smart architecture.

Once removed from a smart architecture, a server configuration can be managed directly.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(vdns_id || vdns_name) && ((dns_id || dns_name || hostaddr))

Input Parameters

vdns_id

The database identifier (ID) of the DNS smart architecture, a unique numeric key value automatically incremented when you add a DNS smart architecture. Use the ID to specify in which smart architecture you add the DNS server.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vdns_name

The name of the DNS smart architecture in which you add the DNS server.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| Type | IPv4 address | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_role

The role of the DNS server within the smart architecture. Note that EfficientIP, EfficientIP package and Microsoft servers can assume any role; BIND/Unbound hybrid servers can only be defined as master or slave; Generic servers can only be defined as master; Amazon Route 53 and Azure servers can only be defined as master or pseudo-master.

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: master slave hidden-master pseudo-master | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Chapter 34. DNS View

Name

dns_view_add — Add/Edit a view

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dnsview_name && (dns_id || dns_name || hostaddr))
- **Editing:** (dnsview_id || (dnsview_name && (dns_id || dns_name || hostaddr)))

Input Parameters

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_addr

Deprecated, replaced by **hostaddr**.

dnsview_order

The level of the DNS view, where 0 represents the highest level in the views hierarchy. The parameters **dnsview_match_client** and **dnsview_match_to** of each view in a server are reviewed following this order.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dnsview_name

The name of the DNS view, each DNS view must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 63 |
| Default value | N/A | Can be edited | Yes |

dnsview_match_clients

The ACL values associated with the match clients configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

| | | | |
|---------------|--------------------------------|----------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dnsview_match_to

The ACL values associated with the match destination configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

| | | | |
|---------------|--------------------------------|----------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dnsview_allow_transfer

The ACL values associated with the allow-transfer configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

| | | | |
|---------------|--------------------------------|----------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_allow_query

The ACL values associated with the allow-query configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

| | | | |
|---------------|--------------------------------|----------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_allow_recursion

The ACL values associated with the allow-recursion configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

| | | | |
|---------------|--------------------------------|----------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_recursion

The recursion status of the DNS view:

Table 34.1. dnsview_recursion possible values

| Status | Description | | |
|---------------|--|----------------|-----|
| no | The view only provides iterative query behavior - normally resulting in a referral. If the answer to the query already exists in the cache it will be returned whatever the value of this statement. | | |
| yes | The view always provides recursive query behavior if requested by the client. | | |
| Type | Fixed value: yes no | Maximum length | N/A |
| Default value | yes | Can be edited | Yes |

dnsview_id

The database identifier (ID) of the DNS view, a unique numeric key value automatically incremented when you add a DNS view. Use the ID to specify the DNS view of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnsview_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | | Can be edited | Yes |

dnsview_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
|---------------|--|----------------|-----|
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnsview_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: `<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....`. If the inheritance or propagation property is not specified, its default value - `set, propagate` - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the view.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dns_view_list — List the views

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dnsview_order

The level of the DNS view, where 0 represents the highest level in the views hierarchy. The parameters **dnsview_match_client** and **dnsview_match_to** of each view in a server are reviewed following this order.

dnsview_recursion

The recursion status of the DNS view:

Table 34.2. *dnsview_recursion* possible values

| Status | Description |
|--------|--|
| no | The view only provides iterative query behavior - normally resulting in a referral. If the answer to the query already exists in the cache it will be returned whatever the value of this statement. |
| yes | The view always provides recursive query behavior if requested by the client. |

dnsview_allow_recursion

The ACL values associated with the allow-recursion configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnsview_allow_query

The ACL values associated with the allow-query configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnsview_allow_transfer

The ACL values associated with the allow-transfer configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnsview_key_name

The name of the DNS TSIG key associated with the DNS view.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

dns_type

The type of the DNS server the object belongs to.

Table 34.3. dns_type possible values

| Type | Description |
|----------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msdaemon | Microsoft Windows DNS server |
| aws | Amazon Route 53 server |
| other | Generic DNS server |
| vdns | EfficientIP DNS smart architecture |

dns_cloud

Internal use. Not documented.

dns_name

The name of the DNS server the object belongs to.

vdns_parent_id

The database identifier (ID) of the DNS smart architecture managing the DNS server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dnsview_name

The name of the DNS view.

dnsview_match_clients

The ACL values associated with the match clients configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnsview_match_to

The ACL values associated with the match destination configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnsview_id

The database identifier (ID) of the DNS view.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

dnsview_class_name

The name of the class applied to the DNS view, it can be preceded by the class directory.

dns_class_name

The name of the class applied to the DNS server the object belongs to, it can be preceded by the class directory.

dns_comment

The description of the DNS server the object belongs to.

gss_keytab_id

The database identifier (ID) of the DNS GSS-TSIG keytab.

dns_version

The version details of the DNS server the object belongs to.

vdns_parent_name

The name of the DNS smart architecture managing the DNS server the object belongs to. # indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

ip_addr

The IPv4 address of the DNS server the object belongs to, in hexadecimal format.

ip6_addr

The IPv6 address of the DNS server the object belongs to, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 34.4. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dnsview_class_parameters

The class parameters applied to the DNS view and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dnsview_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dnsview_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dnsview_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dns_class_parameters

The class parameters applied to the DNS server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dns_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dns_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dns_view_info — Display the properties of a view

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dnsview_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

dnsview_id

The database identifier (ID) of the DNS view. Use the ID to specify the DNS view of your choice.

Output Parameters

dnsview_order

The level of the DNS view, where `0` represents the highest level in the views hierarchy. The parameters `dnsview_match_client` and `dnsview_match_to` of each view in a server are reviewed following this order.

dnsview_recursion

The recursion status of the DNS view:

Table 34.5. `dnsview_recursion` possible values

| Status | Description |
|--------|--|
| no | The view only provides iterative query behavior - normally resulting in a referral. If the answer to the query already exists in the cache it will be returned whatever the value of this statement. |

| Status | Description |
|--------|---|
| yes | The view always provides recursive query behavior if requested by the client. |

dnsview_allow_recursion

The ACL values associated with the allow-recursion configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnsview_allow_query

The ACL values associated with the allow-query configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnsview_allow_transfer

The ACL values associated with the allow-transfer configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnsview_key_name

The name of the DNS TSIG key associated with the DNS view.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

dns_type

The type of the DNS server the object belongs to.

Table 34.6. dns_type possible values

| Type | Description |
|----------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msdaemon | Microsoft Windows DNS server |
| aws | Amazon Route 53 server |
| other | Generic DNS server |
| vdns | EfficientIP DNS smart architecture |

dns_cloud

Internal use. Not documented.

dns_name

The name of the DNS server the object belongs to.

vdns_parent_id

The database identifier (ID) of the DNS smart architecture managing the DNS server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dnsview_name

The name of the DNS view.

dnsview_match_clients

The ACL values associated with the match clients configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnsview_match_to

The ACL values associated with the match destination configuration of the DNS view, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnsview_id

The database identifier (ID) of the DNS view.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

dnsview_class_name

The name of the class applied to the DNS view, it can be preceded by the class directory.

dns_class_name

The name of the class applied to the DNS server the object belongs to, it can be preceded by the class directory.

dns_comment

The description of the DNS server the object belongs to.

gss_keytab_id

The database identifier (ID) of the DNS GSS-TSIG keytab.

dns_version

The version details of the DNS server the object belongs to.

vdns_parent_name

The name of the DNS smart architecture managing the DNS server the object belongs to. # indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

ip_addr

The IPv4 address of the DNS server the object belongs to, in hexadecimal format.

ip6_addr

The IPv6 address of the DNS server the object belongs to, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 34.7. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dnsview_class_parameters

The class parameters applied to the DNS view and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dnsview_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dnsview_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dnsview_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dns_class_parameters

The class parameters applied to the DNS server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dns_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dns_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

dns_view_count — Count the number of views

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

group_dnsview_add — Add a view to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (dnsview_id || (dnsview_name && (dns_id || dns_name || hostaddr))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_id

The database identifier (ID) of the DNS view. Use the ID to specify the DNS view of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_name

The name of the DNS view.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the view.

Name

group_dnsview_delete — Remove a view from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (dnsview_id || (dnsview_name && (dns_id || dns_name || hostaddr))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_id

The database identifier (ID) of the DNS view. Use the ID to specify the DNS view of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnsview_name

The name of the DNS view.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the view.

Name

dns_view_delete — Delete a view

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dnsview_id || (dnsview_name && (dns_id || dns_name || hostaddr)))

Input Parameters

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnsview_id

The database identifier (ID) of the DNS view. Use the ID to specify the DNS view of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnsview_name

The name of the DNS view.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the view.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dns_view_param_add — Add/Edit a DNS option on a view

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dnsview_id && param_key)
- **Editing:** (dnsview_id && param_key)

Input Parameters

dnsview_id

The database identifier (ID) of the DNS view. Use the ID to specify the DNS view of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

param_key

The name of the DNS option you want to add, edit or remove from the view. You can only set one option at a time.

- To add or edit an option: specify its name in the parameter *param_key*, as follows *param_key=<option-name>*, and then specify its value in the parameter *param_value*.
- To remove an option, specify its name in the parameter *param_key* and leave the parameter *param_value* empty.

To set several options, specify as many parameters (*param_key* and *param_value*) as you need.

| Type | String | Maximum length | 64 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

is_array

A way to determine if the DNS view option is an array (1).

| Type | Boolean: 0 1 | Maximum length | N/A |
|---------------|-----------------|----------------|-----|
| Default value | 0 | Can be edited | Yes |

param_value

The value of the DNS option specified in the input *param_key*.

- To add or edit an option value, specify its name in the parameter *param_key* and set its value as follows: *param_value=<option-value>*.

- To remove an option value, specify its name in the parameter *param_key* and leave *param_value* empty: *param_value=*.

| | | | |
|---------------|--------|----------------|--------|
| Type | String | Maximum length | 200000 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- Error*: the service cannot be executed.
- Warning*: the service execution can continue but an issue might have occurred.
- Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the view.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dns_view_param_list — List the DNS options of a view

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

oid

The database identifier (ID) of the DNS option set on the view.

dnsview_id

The database identifier (ID) of the DNS view.

dns_name

The name of the DNS server the object belongs to.

dnsview_name

The name of the DNS view.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

param_key

The name of the DNS option set on the view.

param_value

The value of the DNS option set on the view.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
 - 1 indicates the object is enabled and managed.
 - 2 indicates the object is unmanaged, disabled or both depending on the context.
- By default, *row_enabled* is set to 1 when an object is created.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

read_only

Internal use. Not documented.

Name

dns_view_param_info — Display the properties of a DNS option set on a view

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dnsview_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dnsview_id

The database identifier (ID) of the DNS view. Use the ID to specify the DNS view of your choice.

Output Parameters

oid

The database identifier (ID) of the DNS option set on the view.

dnsview_id

The database identifier (ID) of the DNS view.

dns_name

The name of the DNS server the object belongs to.

dnsview_name

The name of the DNS view.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

param_key

The name of the DNS option set on the view.

param_value

The value of the DNS option set on the view.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

read_only

Internal use. Not documented.

Name

dns_view_param_count — Count the number of DNS options of a view

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dns_view_param_delete — Delete a DNS option from a view

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dnsview_id && param_key)

Input Parameters

dnsview_id

The database identifier (ID) of the DNS view. Use the ID to specify the DNS view of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

param_key

The name of the DNS option that you want to remove from the view: *param_key=<option-name>*.

| Type | String | Maximum length | 64 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the view.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 35. DNS Zone

Name

dns_zone_add — Add/Edit a zone

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dnszone_name && dnszone_type && (dnsview_id || dnsview_name) && (dns_id || dns_name || hostaddr))
- **Editing:** (dnszone_id || (dnszone_name && (dnsview_id || dnsview_name) && (dns_id || dns_name || hostaddr)))

Input Parameters

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

dnszone_site_id

The database identifier (ID) of the space associated with the DNS zone the record belongs to.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dnszone_site_name

The name of the space associated with the DNS zone the record belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_space_name

Deprecated, replaced by **dnszone_site_name**.

dnsview_id

The database identifier (ID) of the DNS view the object belongs to.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dnszone_id

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_name

The name of the DNS view the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

zone

Deprecated, replaced by **dnszone_name**.

dnszone_name

The name of the DNS zone, each DNS zone must have a unique name. For hint zones (**dnszone_type: hint**), you must type in . (dot) as *dnszone_name*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | No |

zone_type

Deprecated, replaced by **dnszone_type**.

dnszone_type

The type of the DNS zone, either *master*, *slave*, *forward*, *stub*, *hint* or *delegation-only*.

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: master slave hint stub forward delegation-only | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

master_addr

Deprecated, replaced by **dnszone_masters**.

dnszone_masters

For slave DNS zones, the IP address of the DNS server and, if relevant, the name of the DNS view that contain the master DNS zone, as follows: *<ip_addr>*; or *<ip_addr> key <dnsview_name>*; .

| | | | |
|------|--------------------------------|----------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
|------|--------------------------------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

forwarders_addr

Deprecated, replaced by **dnszone_forwarders**.

dnszone_forwarders

The IP address(es) of the forwarder(s) associated with the DNS zone. It lists the DNS servers to which any unknown query on this zone should be sent, as follows: <ip_address1>;<ip_address2>;... .

| | | | |
|----------------------|--------------------------------|-----------------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | | Can be edited | Yes |

forward

Deprecated, replaced by **dnszone_forward**.

dnszone_forward

The forwarding mode of the DNS zone.

Table 35.1. dnszone_forward possible values

| Status | Description |
|---------------|--|
| first | The zone sends the queries to the forwarder(s). If no answer is returned, it attempts to answer the queries on its own. |
| only | The zone only forwards the queries to the forwarder(s). Required by some reverse forward zones (e.g., in the case of private addresses). |

If the parameter has no value, it indicates that the forwarding is disabled.

| | | | |
|----------------------|---|-----------------------|-----|
| Type | Fixed value: none first only default | Maximum length | N/A |
| Default value | | Can be edited | Yes |

allow_transfer

Deprecated, replaced by **dnszone_allow_transfer**.

dnszone_allow_transfer

The ACL values associated with the allow-transfer configuration of the DNS zone, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

| | | | |
|----------------------|--------------------------------|-----------------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | | Can be edited | Yes |

allow_query

Deprecated, replaced by **dnszone_allow_query**.

dnszone_allow_query

The ACL values associated with the allow-query configuration of the DNS zone, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

| | | | |
|----------------------|--------------------------------|-----------------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | | Can be edited | Yes |

allow_update

Deprecated, replaced by **dnszone_allow_update**.

dnszone_allow_update

The ACL values associated with the allow-update configuration of the DNS zone, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

| | | | |
|----------------------|--------------------------------|-----------------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | | Can be edited | Yes |

also_notify

Deprecated, replaced by **dnszone_also_notify**.

dnszone_also_notify

The IP address and port of the DNS server managing the smart architecture the DNS zone belongs to. If the parameter **dnszone_notify** is set to yes or explicit, the server specified is instantly notified of any slave zones updates.

| | | | |
|----------------------|--------------------------------|-----------------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | | Can be edited | Yes |

notify

Deprecated, replaced by **dnszone_notify**.

dnszone_notify

The notify status of the DNS zone.

Table 35.2. *dnszone_notify* possible values

| Status | Description |
|----------|--|
| no | No notify message is sent. |
| yes | A notify message is sent to the name servers defined in the NS records of the zone and to the IP address(es) specified in the parameter dnszone_also_notify . |
| explicit | A notify message is sent only to the IP address(es) specified in the parameter dnszone_also_notify . |

The notify message is not sent to the server itself or to the primary server defined in the SOA record of the zone.

| | | | |
|----------------------|------------------------------------|-----------------------|-----|
| Type | Fixed value: yes no explicit | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dnszone_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

dnszone_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

dnszone_ad_integrated

The AD integrated status of the DNS zone. Set it to *1* to indicate that the DNS zone belongs to an Active Directory integrated Microsoft Windows DNS server.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dnszone_is_rpz

The RPZ status of the DNS zone. Set it to *1* to indicate that the DNS zone is a Response Policy Zone.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dnszone_response_policy

The response policy of the DNS zone. All the zones can be set with the policy *given*, only RPZ zones can be set with other policies.

Table 35.3. dnszone_response_policy possible values

| Policy | Description |
|---------------------|---|
| given | All the rules specified in the RPZ zone are applied normally. |
| disabled | The RPZ zone rules configuration is not applied. All the rules it contains are ignored. |
| passthru | The rules specified in the RPZ matching the listed RR names are ignored, no matter the RPZ zone they belong to. |
| nxdomain | The rules specified in the RPZ return an NXDOMAIN response. |
| nodata | The rules specified in the RPZ return a NODATA response. |
| cname <domain-name> | All the rules specified in the RPZ are redirected toward the specified domain name. |

You can only add RPZ zones on EfficientIP or BIND DNS servers.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | given | Can be edited | Yes |

dnszone_rpz_log

The RPZ logging status, if **dnszone_is_rpz** is set to *1* or yes. It allows you to log all the operations triggered by the RPZ rules of the zone.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

dnszone_rpz_recursive_only

A way to *Enable recursive-only on the server* for the zone. By default it is enabled (*1*) and, for this zone, the server only processes policies on recursive queries. It overrides the same option set at server level.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | | | |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

dnszone_rpz_max_policy_ttl

A way to set the number of seconds of the max policy Time To Live of the zone. By default, the parameter is empty, and the *Server max policy TTL* of the zone is of 5 seconds. It overrides the same option set at server level.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

row_enabled

The object activation status.

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.
- If set to 2, the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

use_update_policy

The update policy status of the DNS zone. Set it to 1 to indicate that the DNS zone uses a specific GSS-TSIG/update-policy. You can only configure the zone update policy if the parameter **gss_enabled** is set to 1.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dnszone_order

The level of the RPZ zone, where 0 represents the highest level in the views hierarchy. The RPZ rules of each zone are reviewed following this order. For non-RPZ zones, that have their parameter **dnszone_is_rpz** set to 0, you do not need to set this parameter.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

ddns_scavenging

The DDNS scavenging status of the zone. Set it to 1 to enable the scavenging and automatically delete the stale resource records dynamically added via GSS-TSIG. DDNS scavenging is only effective if the parameters **use_update_policy** and **gss_enabled** are set to 1, and the rule 416 is enabled in the GUI.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>.class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the zone.

ret_oid

The database identifier (ID) of the object you added or edited.

Example

In the example below, we call the service **dns_zone_add** with Ruby (NET::Http) to add a Master zone named *mydomain.tld* in a DNS server that does not contain views.

Example 35.1. Calling the service dns_zone_add using Ruby

```
require 'uri'
require 'net/http'

url = URI("https://solid.intranet/rest/dns_zone_add?" +
          "dnszone_name=mydomain.tld&dnszone_type=master&dns_id=19")

http = Net::HTTP.new(url.host, url.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE

request = Net::HTTP::Post.new(url)
request["x-ipm-username"] = 'aXBtYWRtaW4='
request["x-ipm-password"] = 'YWRtaW4='
request["cache-control"] = 'no-cache'

response = http.request(request)
puts response.read_body
```

Name

dns_zone_list — List the DNS options of a zone

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

num_keys

The number of keys associated with the zone. This number of keys includes all ZSK and KSK.

ipmdns_protocol

Internal use. Not documented.

ipmdns_type

The engine type of the DNS server the DNS zone belongs to: *named* (BIND engine), *nsd* (NSD engine) or *unbound* (Unbound engine).

dns_force_hybrid

Internal use. Not documented.

gss_enabled

The GSS-TSIG status of the DNS server the zone belongs to. 1 indicates that GSS-TSIG is enabled on the server.

gss_keytab_id

The database identifier (ID) of the DNS GSS-TSIG keytab.

use_update_policy

The update policy status of the DNS zone. 1 indicates that the DNS zone uses a specific GSS-TSIG/update-policy. The parameter **gss_enabled** must be set to 1.

dnszone_synching

The synchronization status of the DNS zone. 1 indicates that the zone is currently being synchronized.

dns_state

The status of the DNS server the object belongs to.

Table 35.4. *dns_state* possible values

| Status | Description |
|--------|---|
| ER | The license used in SOLIDserver is not compliant with the added server: the license is invalid. |
| ES | The server configuration could not be parsed properly. |
| ET | The server does not answer anymore due to a scheduled configuration of the server. |
| IC | The SSL credentials are invalid |
| IP | The provided account does not have sufficient privileges to remotely manage the MS server. |

| Status | Description |
|--------|---|
| IR | SOLIDserver cannot resolve the AWS DNS service. The Amazon services are unreachable and the Amazon Route 53 server cannot be managed. Make sure that the DNS resolvers declared on the page <i>Network configuration</i> are valid. |
| IS | There was a setting error during the server declaration. For instance, some settings were added to a server that does not support them or a smart architecture is not managing any physical server. |
| IT | The server editing performed from the GUI is not pushed to the server because SOLIDserver time and date are incorrect. You must use the UTC system on the appliance, especially when managing Amazon Route 53 servers. |
| LS | The server configuration is not viable. |
| N | The server does not have a status as it has not synchronized yet. |
| UE | An error occurred that SOLIDserver could not identify. |
| Y | The server is operational. |

vdns_parent_id

The database identifier (ID) of the DNS smart architecture managing the DNS server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dnszone_allow_update

The ACL values associated with the allow-update configuration of the DNS zone, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnszone_allow_query

The ACL values associated with the allow-query configuration of the DNS zone, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnszone_allow_transfer

The ACL values associated with the allow-transfer configuration of the DNS zone, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnszone_forwarders

The IP address(es) of the forwarder(s) associated with the DNS zone. It lists the DNS servers to which any unknown query on this zone should be sent, as follows: <ip_address1>;<ip_address2>;... .

dnszone_forward

The forwarding mode of the DNS zone.

Table 35.5. dnszone_forward possible values

| Status | Description |
|--------|--|
| first | The zone sends the queries to the forwarder(s). If no answer is returned, it attempts to answer the queries on its own. |
| only | The zone only forwards the queries to the forwarder(s). Required by some reverse forward zones (e.g., in the case of private addresses). |

If the parameter has no value, it indicates that the forwarding is disabled.

dnszone_notify

The notify status of the DNS zone.

Table 35.6. dnszone_notify possible values

| Status | Description |
|----------|--|
| no | No notify message is sent. |
| yes | A notify message is sent to the name servers defined in the NS records of the zone and to the IP address(es) specified in the parameter dnszone_also_notify . |
| explicit | A notify message is sent only to the IP address(es) specified in the parameter dnszone_also_notify . |

The notify message is not sent to the server itself or to the primary server defined in the SOA record of the zone.

dnszone_also_notify

The IP address and port of the DNS server managing the smart architecture the DNS zone belongs to. If the parameter **dnszone_notify** is set to *yes* or *explicit*, the server specified is instantly notified of any slave zones updates.

dnszone_name_utf

The name of the DNS zone in UTF-8 format.

dnszone_id

The database identifier (ID) of the DNS zone.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

dns_type

The type of the DNS server the object belongs to.

Table 35.7. dns_type possible values

| Type | Description |
|----------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msdaemon | Microsoft Windows DNS server |
| aws | Amazon Route 53 server |
| other | Generic DNS server |
| vdns | EfficientIP DNS smart architecture |

dns_cloud

Internal use. Not documented.

dnszone_ad_integrated

The AD integrated status of the DNS zone. 1 indicates that the DNS zone belongs to an Active Directory integrated Microsoft Windows DNS server.

dnszone_sort_zone

Internal use. Not documented.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

dns_name

The name of the DNS server the object belongs to.

dnszone_name

The name of the DNS zone.

dnszone_rev_sort_zone

Internal use. Not documented.

dnszone_is_rpz

The RPZ status of the DNS zone. 1 indicates that the DNS zone is a Response Policy Zone.

dnszone_response_policy

The Overriding rule of the RPZ zone. DNS zones are set with the policy *given*.

dnszone_rpz_log

The logging status of an RPZ zone.

dnszone_rpz_recursive_only

The configuration of the option *Enable recursive-only on the server* of the zone. It overrides the value set at server level. By default it is set to 1 (yes), the server only processes policies on recursive queries.

dnszone_rpz_max_policy_ttl

The configuration of the option *Server max policy TTL*, i.e. the number of seconds of the max policy Time To Live of the zone. It overrides the value set at server level. By default it is empty, i.e. set to 5 seconds.

dnszone_type

The type of the DNS zone, either *master*, *slave*, *forward*, *stub*, *hint* or *delegation-only*.

dnszone_masters

For slave DNS zones, the IP address of the DNS server and, if relevant, the name of the DNS view that contain the master DNS zone, as follows: <ip_addr>; or <ip_addr> key <dnsview_name>;.

dnszone_xfer_done

Internal use. Not documented.

dnszone_is_reverse

A way to determine if the DNS zone provides reverse resolution (1) or direct/name resolution (0).

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

dnszone_order

The level of the DNS zone, where 0 represents the highest level in the zones hierarchy. The RPZ rules parameters of each zone are reviewed following this order. The zones with the parameter dnszone_is_rpz set to 0 will always return 0 for the parameter dnszone_order.

dnszone_site_name

The name of the space associated with the DNS zone.

dnszone_site_id

The database identifier (ID) of the space associated with the DNS zone.

dnszone_class_name

The name of the class applied to the DNS zone, it can be preceded by the class directory.

ddns_scavenging

The DDNS scavenging status of the zone, either enabled (1) or disabled (0). DDNS scavenging is only effective if the parameters **use_update_policy** and **gss_enabled** are set to 1, and the rule 416 is enabled in the GUI.

dnsview_name

The name of the DNS view the object belongs to.

dnsview_id

The database identifier (ID) of the DNS view the object belongs to.

dnsview_class_name

The name of the class applied to the DNS view the object belongs to, it can be preceded by the class directory.

dns_class_name

The name of the class applied to the DNS server the object belongs to, it can be preceded by the class directory.

dns_comment

The description of the DNS server the object belongs to.

dns_version

The version details of the DNS server the object belongs to.

vdns_parent_name

The name of the DNS smart architecture managing the DNS server the object belongs to. # indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

ds

The DNSSEC delegation signer (DS) fingerprint key associated with the DNS zone, if it is signed.

ip_addr

The IPv4 address of the DNS server the object belongs to, in hexadecimal format.

ip6_addr

The IPv6 address of the DNS server the object belongs to, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

dns_vpc_list

The list of cloud networks configured on the DNS private server the zone belongs to, separated by a comma:

- For an Azure server, it returns the list of virtual networks.
- For an Amazon Route 53 server, it returns the list of Virtual Private Cloud (VPC).

aws_delegation_set

The reusable delegation set ID configured on the Amazon Route 53 public server the zone belongs to.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 35.8. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dnszone_class_parameters

The class parameters applied to the DNS zone and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

dnszone_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dnszone_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

dnszone_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

dnsview_class_parameters

The class parameters applied to the DNS view the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

dnsview_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dnsview_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

dns_class_parameters

The class parameters applied to the DNS server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

dns_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dns_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

Example

In the example below, we call the service **dns_zone_list** with Python (Requests) using the clause **WHERE** to retrieve the list of non-RPZ zones that contain .com in their name and belong to a DNS server currently operational and the clause **ORDERBY** to sort them by zone name.

Example 35.2. Calling the service dns_zone_list using Python

```
import requests

url = "https://solid.intranet/rest/dns_zone_list"

querystring = {"WHERE": "dnszone_is_rpz='0' and dns_state='Y' and dnszone_name like '%.com'"}
```

```
'%.tld' , "ORDERBY": "dnszone_name" }

headers = {
    'x-ipm-username': "aXBtYWRtaW4=",
    'x-ipm-password': "YWRtaW4=",
    'cache-control': "no-cache"
}

response = requests.request("GET", url, headers=headers, params=querystring)

print(response.text)
```

Name

dns_zone_info — Display the properties of a zone

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

`dnszone_id`

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

dnszone_id

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

Output Parameters

num_keys

The number of keys associated with the zone. This number of keys includes all ZSK and KSK.

ipmdns_protocol

Internal use. Not documented.

ipmdns_type

The engine type of the DNS server the DNS zone belongs to: `named` (BIND engine), `nsd` (NSD engine) or `unbound` (Unbound engine).

dns_force_hybrid

Internal use. Not documented.

gss_enabled

The GSS-TSIG status of the DNS server the zone belongs to. 1 indicates that GSS-TSIG is enabled on the server.

gss_keytab_id

The database identifier (ID) of the DNS GSS-TSIG keytab.

use_update_policy

The update policy status of the DNS zone. 1 indicates that the DNS zone uses a specific GSS-TSIG/update-policy. The parameter **gss_enabled** must be set to 1.

dnszone_synching

The synchronization status of the DNS zone. 1 indicates that the zone is currently being synchronized.

dns_state

The status of the DNS server the object belongs to.

Table 35.9. dns_state possible values

| Status | Description |
|--------|---|
| ER | The license used in SOLIDserver is not compliant with the added server: the license is invalid. |
| ES | The server configuration could not be parsed properly. |
| ET | The server does not answer anymore due to a scheduled configuration of the server. |
| IC | The SSL credentials are invalid |
| IP | The provided account does not have sufficient privileges to remotely manage the MS server. |
| IR | SOLIDserver cannot resolve the AWS DNS service. The Amazon services are unreachable and the Amazon Route 53 server cannot be managed. Make sure that the DNS resolvers declared on the page <i>Network configuration</i> are valid. |
| IS | There was a setting error during the server declaration. For instance, some settings were added to a server that does not support them or a smart architecture is not managing any physical server. |
| IT | The server editing performed from the GUI is not pushed to the server because SOLIDserver time and date are incorrect. You must use the UTC system on the appliance, especially when managing Amazon Route 53 servers. |
| LS | The server configuration is not viable. |
| N | The server does not have a status as it has not synchronized yet. |
| UE | An error occurred that SOLIDserver could not identify. |
| Y | The server is operational. |

vdns_parent_id

The database identifier (ID) of the DNS smart architecture managing the DNS server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dnszone_allow_update

The ACL values associated with the allow-update configuration of the DNS zone, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnszone_allow_query

The ACL values associated with the allow-query configuration of the DNS zone, as follows: <value1>;<value2>;... . Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnszone_allow_transfer

The ACL values associated with the allow-transfer configuration of the DNS zone, as follows: <value1>;<value2>.... Values may include IP and network addresses, the name of TSIG keys and ACLs, preceded by ! if the access is denied.

dnszone_forwarders

The IP address(es) of the forwarder(s) associated with the DNS zone. It lists the DNS servers to which any unknown query on this zone should be sent, as follows: <ip_address1>;<ip_address2>....

dnszone_forward

The forwarding mode of the DNS zone.

Table 35.10. dnszone_forward possible values

| Status | Description |
|--------|--|
| first | The zone sends the queries to the forwarder(s). If no answer is returned, it attempts to answer the queries on its own. |
| only | The zone only forwards the queries to the forwarder(s). Required by some reverse forward zones (e.g., in the case of private addresses). |

If the parameter has no value, it indicates that the forwarding is disabled.

dnszone_notify

The notify status of the DNS zone.

Table 35.11. dnszone_notify possible values

| Status | Description |
|----------|--|
| no | No notify message is sent. |
| yes | A notify message is sent to the name servers defined in the NS records of the zone and to the IP address(es) specified in the parameter dnszone_also_notify . |
| explicit | A notify message is sent only to the IP address(es) specified in the parameter dnszone_also_notify . |

The notify message is not sent to the server itself or to the primary server defined in the SOA record of the zone.

dnszone_also_notify

The IP address and port of the DNS server managing the smart architecture the DNS zone belongs to. If the parameter **dnszone_notify** is set to yes or explicit, the server specified is instantly notified of any slave zones updates.

dnszone_name_utf

The name of the DNS zone in UTF-8 format.

dnszone_id

The database identifier (ID) of the DNS zone.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

dns_type

The type of the DNS server the object belongs to.

Table 35.12. dns_type possible values

| Type | Description |
|----------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msdaemon | Microsoft Windows DNS server |
| aws | Amazon Route 53 server |
| other | Generic DNS server |
| vdns | EfficientIP DNS smart architecture |

dns_cloud

Internal use. Not documented.

dnszone_ad_integrated

The AD integrated status of the DNS zone. 1 indicates that the DNS zone belongs to an Active Directory integrated Microsoft Windows DNS server.

dnszone_sort_zone

Internal use. Not documented.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

dns_name

The name of the DNS server the object belongs to.

dnszone_name

The name of the DNS zone.

dnszone_rev_sort_zone

Internal use. Not documented.

dnszone_is_rpz

The RPZ status of the DNS zone. 1 indicates that the DNS zone is a Response Policy Zone.

dnszone_response_policy

The Overriding rule of the RPZ zone. DNS zones are set with the policy *given*.

dnszone_rpz_log

The logging status of an RPZ zone.

dnszone_rpz_recursive_only

The configuration of the option *Enable recursive-only on the server* of the zone. It overrides the value set at server level. By default it is set to 1 (yes), the server only processes policies on recursive queries.

dnszone_rpz_max_policy_ttl

The configuration of the option *Server max policy TTL*, i.e. the number of seconds of the max policy Time To Live of the zone. It overrides the value set at server level. By default it is empty, i.e. set to 5 seconds.

dnszone_type

The type of the DNS zone, either *master*, *slave*, *forward*, *stub*, *hint* or *delegation-only*.

dnszone_masters

For slave DNS zones, the IP address of the DNS server and, if relevant, the name of the DNS view that contain the master DNS zone, as follows: <ip_addr>; or <ip_addr> key <dnsview_name>; .

dnszone_xfer_done

Internal use. Not documented.

dnszone_is_reverse

A way to determine if the DNS zone provides reverse resolution (1) or direct/name resolution (0),

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

dnszone_order

The level of the DNS zone, where 0 represents the highest level in the zones hierarchy. The RPZ rules parameters of each zone are reviewed following this order. The zones with the parameter dnszone_is_rpz set to 0 will always return 0 for the parameter dnszone_order.

dnszone_site_name

The name of the space associated with the DNS zone.

dnszone_site_id

The database identifier (ID) of the space associated with the DNS zone.

dnszone_class_name

The name of the class applied to the DNS zone, it can be preceded by the class directory.

ddns_scavenging

The DDNS scavenging status of the zone, either enabled (1) or disabled (0). DDNS scavenging is only effective if the parameters **use_update_policy** and **gss_enabled** are set to 1, and the rule 416 is enabled in the GUI.

dnsview_name

The name of the DNS view the object belongs to.

dnsview_id

The database identifier (ID) of the DNS view the object belongs to.

dnsview_class_name

The name of the class applied to the DNS view the object belongs to, it can be preceded by the class directory.

dns_class_name

The name of the class applied to the DNS server the object belongs to, it can be preceded by the class directory.

dns_comment

The description of the DNS server the object belongs to.

dns_version

The version details of the DNS server the object belongs to.

vdns_parent_name

The name of the DNS smart architecture managing the DNS server the object belongs to. # indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

ds

The DNSSEC delegation signer (DS) fingerprint key associated with the DNS zone, if it is signed.

ip_addr

The IPv4 address of the DNS server the object belongs to, in hexadecimal format.

ip6_addr

The IPv6 address of the DNS server the object belongs to, in hexadecimal format.

hostaddr

The human readable version of the parameter **ip_addr** or **ip6_addr**.

dns_vpc_list

The list of cloud networks configured on the DNS private server the zone belongs to, separated by a comma:

- For an Azure server, it returns the list of virtual networks.
- For an Amazon Route 53 server, it returns the list of Virtual Private Cloud (VPC).

aws_delegation_set

The reusable delegation set ID configured on the Amazon Route 53 public server the zone belongs to.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 35.13. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

dnszone_class_parameters

The class parameters applied to the DNS zone and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dnszone_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dnszone_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dnszone_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dnsview_class_parameters

The class parameters applied to the DNS view the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dnsview_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dnsview_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dns_class_parameters

The class parameters applied to the DNS server the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dns_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dns_class_parameters**: <class-parameter1>=<inheritance>, <propagation>&<class-parameter2>=<inheritance>&.... .

Name

dns_zone_count — Count the number of zones

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dns_zone_groupby — Group zones by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: `GROUPBY=<param1>,<param2>,...`. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement `SELECT` is returned.

Name

dns_zone_groupby_count — Count the number of zones grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

group_dnszone_add — Add a zone to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (dnszone_id || (dnszone_name && (dnsview_id || (dnsview_name && (dns_id || dns_name || hostaddr))))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_id

The database identifier (ID) of the DNS view the object belongs to.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_name

The name of the DNS view the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_id

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_name

The name of the DNS zone the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

zone

Deprecated, replaced by **dnszone_name**.

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the zone.

Name

group_dnszone_delete — Remove a zone from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (dnszone_id || (dnszone_name && (dnsview_id || (dnsview_name && (dns_id || dns_name || hostaddr))))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_id

The database identifier (ID) of the DNS view the object belongs to.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_name

The name of the DNS view the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_id

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_name

The name of the DNS zone the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

zone

Deprecated, replaced by **dnszone_name**.

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the zone.

Name

dns_zone_delete — Delete a zone

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dnszone_id || (dnszone_name && (dnsview_id || (dnsview_name && (dns_id || dns_name || hostaddr))))

Input Parameters

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_id

The database identifier (ID) of the DNS view the object belongs to.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_name

The name of the DNS view the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_id

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_name

The name of the DNS zone the object belongs to.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

zone

Deprecated, replaced by **dnszone_name**.

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the zone.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dns_zone_param_add — Add/Edit a DNS option on a zone

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dnszone_id && param_key)
- **Editing:** (dnszone_id && param_key)

Input Parameters

dnszone_id

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

param_key

The name of the DNS option you want to add, edit or remove from the zone. You can only set one option at a time.

- To add or edit an option: specify its name in the parameter *param_key*, as follows *param_key=<option-name>*, and then specify its value in the parameter *param_value*.
- To remove an option, specify its name in the parameter *param_key* and leave the parameter *param_value* empty.

To set several options, specify as many parameters (*param_key* and *param_value*) as you need.

| Type | String | Maximum length | 64 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

is_array

A way to determine if the DNS zone option is an array (1).

| Type | Boolean: 0 1 | Maximum length | N/A |
|---------------|-----------------|----------------|-----|
| Default value | 0 | Can be edited | Yes |

param_value

The value of the DNS option specified in the input *param_key*.

- To add or edit an option value, specify its name in the parameter *param_key* and set its value as follows: *param_value=<option-value>*.

- To remove an option value, specify its name in the parameter *param_key* and leave *param_value* empty: *param_value=*.

| | | | |
|---------------|--------|----------------|--------|
| Type | String | Maximum length | 200000 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- Error*: the service cannot be executed.
- Warning*: the service execution can continue but an issue might have occurred.
- Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the zone.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dns_zone_param_list — List the DNS options of a zone

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

oid

The database identifier (ID) of the DNS option set on the zone.

dnszone_id

The database identifier (ID) of the DNS zone.

dns_name

The name of the DNS server the object belongs to.

dnszone_name

The name of the DNS zone.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

param_key

The name of the DNS option set on the zone.

param_value

The value of the DNS option set on the zone.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
 - 1 indicates the object is enabled and managed.
 - 2 indicates the object is unmanaged, disabled or both depending on the context.
- By default, *row_enabled* is set to 1 when an object is created.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

read_only

Internal use. Not documented.

Name

dns_zone_param_info — Display the properties of a DNS option set on a zone

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

`dnszone_id`

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

`dnszone_id`

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

Output Parameters

`oid`

The database identifier (ID) of the DNS option set on the zone.

`dnszone_id`

The database identifier (ID) of the DNS zone.

`dns_name`

The name of the DNS server the object belongs to.

`dnszone_name`

The name of the DNS zone.

`dns_id`

The database identifier (ID) of the DNS server the object belongs to.

`param_key`

The name of the DNS option set on the zone.

param_value

The value of the DNS option set on the zone.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

read_only

Internal use. Not documented.

Name

dns_zone_param_count — Count the number of DNS options of a zone

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dns_zone_param_delete — Delete a DNS option from a zone

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dnszone_id && param_key)

Input Parameters

dnszone_id

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

param_key

The name of the DNS option that you want to remove from the zone: *param_key=<option-name>*.

| Type | String | Maximum length | 64 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the DNS option set on the zone.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 36. DNS Resource Record

Name

dns_rr_add — Add/Edit a resource record

Description

This service allows you to add a resource record or edit an existing one.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Adding some resource records requires to specify one or more values:

Table 36.1. Expected values for the DNS records

| Type | Value number | Related field(s) | Description |
|-------|--------------|------------------|---|
| SOA | 1 | Name server | The FQDN of the primary Master name server for the zone the record belongs to. Has a special meaning when used with Dynamic DNS (DDNS): called MNAME, it allows the DNS client to know on which DNS server it has to update itself with DDNS. |
| | 2 | Responsible | The administrator email address for the zone the record belongs to. |
| | 3 | Serial number | The serial number for the zone the record belongs to. The serial number is automatically incremented for each zone change. |
| | 4 | Refresh | The refresh delay for the zone the record belongs to, in seconds. When reached, it forces the slave name server(s) to read the SOA record. If this record is higher than the slave's one, a zone transfer will be triggered by the slave to get the latest version of the zone. Typical values are 3 to 24 hours. |
| | 5 | Retry | The retry delay for the zone the record belongs to, in seconds. When reached, it forces the slave server to retry the request if it fails to reach the master server during a refresh cycle. Typical values are from 10 to 60 minutes. |
| | 6 | Expiration | The expiration time for the zone the record belongs to, in seconds. When reached, the zone records are considered to be no longer valid/authoritative. The DNS server then stops responding to queries for the zone. To avoid a major outage, the typical value is pretty high, between 1 to 3 weeks. |
| | 7 | Minimum | The minimum time for the zone the record belongs to, in seconds. It defines the period of time that negative responses can be cached from the slave. For instance, if a request cannot be resolved, the server will answer with a NXDOMAIN result (No such domain). The server will continue returning this value until the Minimum value expires, then it will retry the resolution. This value has to be between 0 and 3 hours. |
| NS | 1 | DNS server | The DNS server hostname. |
| MX | 1 | Preference | A number, between 0 and 65535, to define which server has priority if there are several RRs in the zone. The lowest the value has the priority over the other server(s). |
| | 2 | Mail server | The SMTP (mail) server hostname. |
| A | 1 | IP address | The IPv4 Address of the host. |
| AAAA | 1 | IPv6 address | The IPv6 Address of the host. |
| PTR | 1 | Localization | The hostname that should be returned when the IP address is queried. |
| CNAME | 1 | Hostname | The hostname. |
| TXT | 1 | Text | The description of your choice (max. 255 characters including spaces). |

| Type | Value number | Related field(s) | Description |
|--|--------------|-------------------|---|
| SRV | 1 | Priority | A number, between 0 and 65535, to define which server has priority if there are several SRV RRs in the zone. The lowest the value has the priority over the other server(s). |
| | 2 | Weight | A number, between 0 and 65535, that defines the server weight. If two SRV RRs have the same priority, the weight defines which server is more used. The field gives priority to the SRV RR with the greatest weight value: the greater the value is, the more the server is solicited. If you type in 0, there is no weighting. |
| | 3 | Ports | The port number that delivers the service to the target. |
| | 4 | Target | The hostname of the server delivering the service. |
| HINFO | 1 | CPU | The name of the CPU, either INTEL, AMD, SPARC, ALPHA, HPPA, POWERPC, MIPS, MOTOROLA or Other. |
| | 2 | OS | The name of the operating system, either AIX, FREEBSD, HPUX, IRIX, LINUX, OSF, OS/2, SOLARIS, SUNOS, VMS, WINDOWS, or Other. |
| MINFO | 1 | Responsible email | The email address of the administrator of the mail list. |
| | 2 | Error email | The email address that should receive the error messages regarding the mail list. |
| DNAME | 1 | Domain | The domain name of a subdomain of the zone. |
| AFSDB | 1 | Preference | Type the version of AFS service used: 1 (AFS version 3.0) or 2 (OSF DCE/NCA version). |
| | 2 | AFS server | The AFS hostname. |
| NAPTR | 1 | Order | A number, between 0 and 65535, to define which RR has priority if there are several NAPTR RRs in the zone. The lowest the value has the priority over the other record(s). |
| | 2 | Preference | A number, between 0 and 65535, to define which RR has priority if there are several NAPTR RRs have the same order in the zone. The lowest the value has the priority over the other record(s). |
| | 2 | Flags | The string that corresponds to the action you want your client application to perform. The flag specified impacts the data expected in the field Services, Regex and/or Replace. |
| | 3 | Services | The services parameters to which applies the action specified in the field Flags. You must respect your client application syntax. |
| | 4 | Regex | The string that contains a substitution expression matching the format <delimit ereg delimit substitution delimit flag> to which applies the action specified the field Flags. |
| | 5 | Replace | An FQDN domain name to which applies the action specified the field Flags. You can specify no domain name if you type in . (dot) in the field. |
| For more details, refer to RFC 3403 available on IETF website at http://tools.ietf.org/html/rfc3403 . | | | |
| NSAP | 1 | Name | The NSAP address of the end system. It should start with 0x and not exceed 255 hexadecimal characters separated by dots. |
| DS | 1 | Key Tag | The parent zone DS key tag. |
| | 2 | Key Algorithm | The parent zone DS algorithm key. |
| | 3 | Digest Type | The parent zone DS digest type. |
| | 4 | Digest | The parent zone DS digest. |
| DNSKEY | 1 | Flags | The zone key flag. |
| | 2 | Protocol | The protocol value. |
| | 3 | Algorithm | The public key's cryptographic algorithm. |
| | 4 | Key | The public key material. |

| Type | Value number | Related field(s) | Description |
|------|--------------|------------------|---|
| WKS | 1 | IP address | The IPv4 Address of the host that contains the services listed in the Services field. |
| | 2 | Protocol | The communication protocol, either <i>TCP</i> or <i>UDP</i> . |
| | 3 | Services | The list of needed services. |

Mandatory Input Parameters

- **Addition:** (rr_name && rr_type && value1 && (dns_id || dns_name || hostaddr))
- **Editing:** (rr_id || (rr_name && rr_type && value1 && (dns_id || dns_name || hostaddr)))

Input Parameters

rr_id

The database identifier (ID) of the DNS resource record, a unique numeric key value automatically incremented when you add a DNS RR. Use the ID to specify the record of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_id

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

zone

Deprecated, replaced by **dnszone_name**.

dnszone_name

The name of the DNS zone the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_site_id

The database identifier (ID) of the space associated with the DNS zone the record belongs to.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_id

The database identifier (ID) of the DNS view, a unique numeric key value automatically incremented when you add a DNS view. Use the ID to specify the DNS view of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_name

The name of the DNS view.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_glue

The shortname of the DNS resource record.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

serial_type

The syntax format to use for the DNS resource record serial number. By default, *rfc1912* is used (YYYYMMDDnn) but you can also specify the *unix_timestamp* format.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | Fixed value: date | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_name

The full name of the DNS resource record. Specify it as value, it can either follow the format <rr-name>.<existing-zone-name>.<extension> or be . (a dot).

| | | | |
|---------------|--|----------------|-----|
| Type | Regular expression: ^([^\.]{1,63})(\.[^\.]{1,63})*\.\?\$_ | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

rr_ttl

The time to live of the DNS resource record, in seconds.

| | | | |
|------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
|------|--------------|----------------|-----|

| | | | |
|----------------------|------|----------------------|-----|
| Default value | 3600 | Can be edited | Yes |
|----------------------|------|----------------------|-----|

rr_type

The type of the DNS resource record.

Table 36.2. rr_type possible values

| Value | Record type description |
|--------|---|
| SOA | Start of Authority. Defines the zone name, an email contact and various time and refresh values applicable to the zone. It is automatically generated upon creation of a zone and cannot be added manually. |
| NS | Name Server. Defines the authoritative name server(s) for the domain (defined by the SOA record) or the subdomain. The NS record that indicates which server has authority over a zone is automatically generated upon the creation of a zone, once the server has been synchronized. |
| A | IPv4 Address. An IPv4 address for a host. |
| PTR | Pointer Record. Address Resolution, from an IP address (IPv4 or IPv6) to a host. Used in reverse mapping. |
| AAAA | IPv6 Address. An IPv6 address for a host. |
| CNAME | Canonical Name. An alias name for a host. |
| MX | Mail Exchange. The mail server/exchanger that services this zone. |
| SRV | Services record. Defines services available in the zone, for example, LDAP, HTTP, etc... |
| DNAME | Delegation of Reverse Names. Delegation of reverse addresses primarily in IPv6. (Deprecated, use the CNAME RR instead) |
| TXT | Text. Information associated with a name. |
| DS | Delegation Signer, a DNSSEC related RR used to verify the validity of the ZSK of a subdomain. |
| DNSKEY | DNS Key. It contains the public cryptographic key used to sign the zone with DNSSEC. |
| 65534 | A private type record automatically added to the zone once it is signed with DNSSEC. |
| HINFO | System Information. Information about a host: hardware type and operating system description. |
| MINFO | Mailbox mail list Information. Defines the mail administrator for a mail list and optionally a mailbox to receive error messages relating to the mail list. |
| AFSDB | AFS Database. Location of the AFS servers. |
| WKS | Well-Known Service. Defines the services and protocols supported by a host. (Deprecated, use the SRV RR instead) |
| NAPTR | Naming Authority Pointer Record. General purpose definition of rule set to be used by applications e.g. VoIP. |
| NSAP | Network Service Access Point. Defines record (equivalent of an A record) maps a host name to an endpoint address. |

Note that the parameter is not case sensitive, you could type in A or a.

| | | | |
|-------------|--|-----------------------|-----|
| Type | Fixed value: A NS MD MF CNAME SOA MB MG MR NULL WKS PTR HINFO MINFO MX TXT SPF RP AFSDB X25 ISDN RT NSAP NSAP_PTR SIG KEY PX GPOS AAAA LOC NXT EID NIMLOC SRV ATMA NAPTR KX CERT A6 DNAME OPT DS DNSSIG NSEC DNSKEY NSEC3 NSEC3PARAM CDS CDNSKEY CAA TLSA SSHFP OPENPGPKEY URI AVC | Maximum length | N/A |
|-------------|--|-----------------------|-----|

| | | | |
|----------------------|--|----------------------|-----|
| | NINFO DLV DHCID EUI48 EUI64 NID L32 L64 HTTPS SVCB | | |
| Default value | N/A | Can be edited | Yes |

rr_value1

Deprecated, replaced by **value1**.

value1

The first or only value required for the DNS resource record, as detailed in the service description.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value2

Deprecated, replaced by **value2**.

value2

The second value of the DNS resource record, depending on its type, as detailed in the service description.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value3

Deprecated, replaced by **value3**.

value3

The third value of the DNS resource record, depending on its type, as detailed in the service description.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value4

Deprecated, replaced by **value4**.

value4

The fourth value of the DNS resource record, depending on its type, as detailed in the service description.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value5

Deprecated, replaced by **value5**.

value5

The fifth value of the DNS resource record, depending on its type, as detailed in the service description.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value6

Deprecated, replaced by **value6**.

value6

The sixth value of the DNS resource record, depending on its type, as detailed in the service description.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value7

Deprecated, replaced by **value7**.

value7

The seventh value of the DNS resource record, depending on its type, as detailed in the service description.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

check_value

A way to check the values of the DNS resource record before upon addition (1) in order to create a record with the same name but with different values.

| | | | |
|---------------|------------------------|----------------|-----|
| Type | Fixed value: yes no | Maximum length | N/A |
| Default value | yes | Can be edited | Yes |

rr_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

rr_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>.class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... . If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.

- *Warning:* the service execution can continue but an issue might have occurred.
- *Notice:* the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Example

In the example below, we call the service **dns_rr_add** with PHP (cURL) to add an A record in the zone *mydomain.tld*.

Example 36.1. Calling the service dns_rr_add using PHP

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/dns_rr_add?" .
        "&rr_name=www.mydomain.tld&rr_type=A&value1=192.168.0.153&dns_id=19",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "POST",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Name

dns_rr_list — List the resource records

Description

This service allows you to list the resource records.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Adding some resource records requires to specify one or more values:

Table 36.3. Expected values for the DNS records

| Type | Value number | Related field(s) | Description |
|-------|--------------|------------------|---|
| SOA | 1 | Name server | The FQDN of the primary Master name server for the zone the record belongs to. Has a special meaning when used with Dynamic DNS (DDNS): called MNAME, it allows the DNS client to know on which DNS server it has to update itself with DDNS. |
| | 2 | Responsible | The administrator email address for the zone the record belongs to. |
| | 3 | Serial number | The serial number for the zone the record belongs to. The serial number is automatically incremented for each zone change. |
| | 4 | Refresh | The refresh delay for the zone the record belongs to, in seconds. When reached, it forces the slave name server(s) to read the SOA record. If this record is higher than the slave's one, a zone transfer will be triggered by the slave to get the latest version of the zone. Typical values are 3 to 24 hours. |
| | 5 | Retry | The retry delay for the zone the record belongs to, in seconds. When reached, it forces the slave server to retry the request if it fails to reach the master server during a refresh cycle. Typical values are from 10 to 60 minutes. |
| | 6 | Expiration | The expiration time for the zone the record belongs to, in seconds. When reached, the zone records are considered to be no longer valid/authoritative. The DNS server then stops responding to queries for the zone. To avoid a major outage, the typical value is pretty high, between 1 to 3 weeks. |
| | 7 | Minimum | The minimum time for the zone the record belongs to, in seconds. It defines the period of time that negative responses can be cached from the slave. For instance, if a request cannot be resolved, the server will answer with a NXDOMAIN result (No such domain). The server will continue returning this value until the Minimum value expires, then it will retry the resolution. This value has to be between 0 and 3 hours. |
| NS | 1 | DNS server | The DNS server hostname. |
| MX | 1 | Preference | A number, between 0 and 65535, to define which server has priority if there are several RRs in the zone. The lowest the value has the priority over the other server(s). |
| | 2 | Mail server | The SMTP (mail) server hostname. |
| A | 1 | IP address | The IPv4 Address of the host. |
| AAAA | 1 | IPv6 address | The IPv6 Address of the host. |
| PTR | 1 | Localization | The hostname that should be returned when the IP address is queried. |
| CNAME | 1 | Hostname | The hostname. |
| TXT | 1 | Text | The description of your choice (max. 255 characters including spaces). |

| Type | Value number | Related field(s) | Description |
|--|--------------|-------------------|---|
| SRV | 1 | Priority | A number, between 0 and 65535, to define which server has priority if there are several SRV RRs in the zone. The lowest the value has the priority over the other server(s). |
| | 2 | Weight | A number, between 0 and 65535, that defines the server weight. If two SRV RRs have the same priority, the weight defines which server is more used. The field gives priority to the SRV RR with the greatest weight value: the greater the value is, the more the server is solicited. If you type in 0, there is no weighting. |
| | 3 | Ports | The port number that delivers the service to the target. |
| | 4 | Target | The hostname of the server delivering the service. |
| HINFO | 1 | CPU | The name of the CPU, either INTEL, AMD, SPARC, ALPHA, HPPA, POWERPC, MIPS, MOTOROLA or Other. |
| | 2 | OS | The name of the operating system, either AIX, FREEBSD, HPUX, IRIX, LINUX, OSF, OS/2, SOLARIS, SUNOS, VMS, WINDOWS, or Other. |
| MINFO | 1 | Responsible email | The email address of the administrator of the mail list. |
| | 2 | Error email | The email address that should receive the error messages regarding the mail list. |
| DNAME | 1 | Domain | The domain name of a subdomain of the zone. |
| AFSDB | 1 | Preference | Type the version of AFS service used: 1 (AFS version 3.0) or 2 (OSF DCE/NCA version). |
| | 2 | AFS server | The AFS hostname. |
| NAPTR | 1 | Order | A number, between 0 and 65535, to define which RR has priority if there are several NAPTR RRs in the zone. The lowest the value has the priority over the other record(s). |
| | 2 | Preference | A number, between 0 and 65535, to define which RR has priority if there are several NAPTR RRs have the same order in the zone. The lowest the value has the priority over the other record(s). |
| | 2 | Flags | The string that corresponds to the action you want your client application to perform. The flag specified impacts the data expected in the field Services, Regex and/or Replace. |
| | 3 | Services | The services parameters to which applies the action specified in the field Flags. You must respect your client application syntax. |
| | 4 | Regex | The string that contains a substitution expression matching the format <delimit ereg delimit substitution delimit flag> to which applies the action specified the field Flags. |
| | 5 | Replace | An FQDN domain name to which applies the action specified the field Flags. You can specify no domain name if you type in . (dot) in the field. |
| For more details, refer to RFC 3403 available on IETF website at http://tools.ietf.org/html/rfc3403 . | | | |
| NSAP | 1 | Name | The NSAP address of the end system. It should start with 0x and not exceed 255 hexadecimal characters separated by dots. |
| DS | 1 | Key Tag | The parent zone DS key tag. |
| | 2 | Key Algorithm | The parent zone DS algorithm key. |
| | 3 | Digest Type | The parent zone DS digest type. |
| | 4 | Digest | The parent zone DS digest. |
| DNSKEY | 1 | Flags | The zone key flag. |
| | 2 | Protocol | The protocol value. |
| | 3 | Algorithm | The public key's cryptographic algorithm. |
| | 4 | Key | The public key material. |

| Type | Value number | Related field(s) | Description |
|------|--------------|------------------|---|
| WKS | 1 | IP address | The IPv4 Address of the host that contains the services listed in the Services field. |
| | 2 | Protocol | The communication protocol, either <i>TCP</i> or <i>UDP</i> . |
| | 3 | Services | The list of needed services. |

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

rr_all_value

The concatenated values of the DNS resource record, as follows: <value1>, <value2>, <value3>, <value4>, <value5>, <value6>, <value7>.

dnszone_sort_zone

Internal use. Not documented.

dnszone_is_rpz

The RPZ status of the DNS zone the resource record belongs to. 1 indicates that the DNS zone the record belongs to is a Response Policy Zone.

dnszone_type

The type of the DNS zone the object belongs to, either *master*, *slave*, *forward*, *stub*, *hint* or *delegation-only*.

rr_full_name

The full name of the DNS resource record.

rr_full_name_utf

The name of the DNS resource record in UTF-8 format.

rr_name_ip_addr

Internal use. Not documented.

rr_name_ip4_addr

Internal use. Not documented.

rr_value_ip_addr

Internal use. Not documented.

rr_value_ip4_addr

Internal use. Not documented.

rr_glue

The shortname of the DNS resource record.

rr_type

The type of the DNS resource record.

Table 36.4. *rr_type* possible values

| Value | Record type description |
|-------|---|
| SOA | Start of Authority. Defines the zone name, an email contact and various time and refresh values applicable to the zone. It is automatically generated upon creation of a zone and cannot be added manually. |
| NS | Name Server. Defines the authoritative name server(s) for the domain (defined by the SOA record) or the subdomain. The NS record that indicates which server has authority over a |

| Value | Record type description |
|--------|---|
| | zone is automatically generated upon the creation of a zone, once the server has been synchronized. |
| A | IPv4 Address. An IPv4 address for a host. |
| PTR | Pointer Record. Address Resolution, from an IP address (IPv4 or IPv6) to a host. Used in reverse mapping. |
| AAAA | IPv6 Address. An IPv6 address for a host. |
| CNAME | Canonical Name. An alias name for a host. |
| MX | Mail Exchange. The mail server/exchanger that services this zone. |
| SRV | Services record. Defines services available in the zone, for example, LDAP, HTTP, etc... |
| DNAME | Delegation of Reverse Names. Delegation of reverse addresses primarily in IPv6. (Deprecated, use the CNAME RR instead) |
| TXT | Text. Information associated with a name. |
| DS | Delegation Signer, a DNSSEC related RR used to verify the validity of the ZSK of a subdomain. |
| DNSKEY | DNS Key. It contains the public cryptographic key used to sign the zone with DNSSEC. |
| 65534 | A private type record automatically added to the zone once it is signed with DNSSEC. |
| HINFO | System Information. Information about a host: hardware type and operating system description. |
| MINFO | Mailbox mail list Information. Defines the mail administrator for a mail list and optionally a mailbox to receive error messages relating to the mail list. |
| AFSDB | AFS Database. Location of the AFS servers. |
| WKS | Well-Known Service. Defines the services and protocols supported by a host. (Deprecated, use the SRV RR instead) |
| NAPTR | Naming Authority Pointer Record. General purpose definition of rule set to be used by applications e.g. VoIP. |
| NSAP | Network Service Access Point. Defines record (equivalent of an A record) maps a host name to an endpoint address. |

ttl

The time to live of the DNS resource record, in seconds.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

rr_class_name

The name of the class applied to the resource record, it can be preceded by the class directory.

value1

The first or only value required for the DNS resource record, as detailed in the service description.

value2

The second value of the DNS resource record, depending on its type, as detailed in the service description.

value3

The third value of the DNS resource record, depending on its type, as detailed in the service description.

value4

The fourth value of the DNS resource record, depending on its type, as detailed in the service description.

value5

The fifth value of the DNS resource record, depending on its type, as detailed in the service description.

value6

The sixth value of the DNS resource record, depending on its type, as detailed in the service description.

value7

The seventh value of the DNS resource record, depending on its type, as detailed in the service description.

dnszone_id

The database identifier (ID) of the DNS zone the object belongs to.

rr_id

The database identifier (ID) of the DNS resource record.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

dnszone_name_utf

The name of the DNS zone the resource record belongs to, in UTF-8 format.

dnszone_name

The name of the DNS zone the object belongs to.

dns_name

The name of the DNS server the object belongs to.

dns_type

The type of the DNS server the object belongs to.

Table 36.5. dns_type possible values

| Type | Description |
|----------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msdaemon | Microsoft Windows DNS server |
| aws | Amazon Route 53 server |
| other | Generic DNS server |
| vdns | EfficientIP DNS smart architecture |

dns_cloud

Internal use. Not documented.

vdns_parent_id

The database identifier (ID) of the DNS smart architecture managing the DNS server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dnsview_name

The name of the DNS view the object belongs to.

dnsview_class_name

The name of the class applied to the DNS view the object belongs to, it can be preceded by the class directory.

dnsview_id

The database identifier (ID) of the DNS view the object belongs to.

dnszone_site_name

The name of the space associated with the DNS zone the RR belongs to.

dnszone_is_reverse

A way to determine if the DNS zone the resource record belongs to provides reverse resolution (1) or direct/name resolution (0),

dnszone_masters

For resource records in slave DNS zones, the IP address of the DNS server and, if relevant, the name of the DNS view that contain the master DNS zone, as follows: <ip_addr>; or <ip_addr> key <dnsview_name>; .

vdns_parent_name

The name of the DNS smart architecture managing the DNS server the object belongs to. # indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dnszone_forwarders

The IP address(es) of the forwarder(s) associated with the DNS zone the resource record belongs to. It lists the DNS servers to which any unknown query on this zone should be sent, as follows: <ip_address1>;<ip_address2>;... .

dns_class_name

The name of the class applied to the DNS server the object belongs to, it can be preceded by the class directory.

dnszone_class_name

The name of the class applied to the DNS zone the object belongs to, it can be preceded by the class directory.

dns_version

The version details of the DNS server the object belongs to.

dns_comment

The description of the DNS server the object belongs to.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 36.6. Multi-status severity levels

| Message number | Severity | Description |
|----------------|-----------|---|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |

| Message number | Severity | Description |
|----------------|---------------|--|
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

rr_auth_gsstsig

A way to determine if the record was authenticated via dynamic update (GSS-TSIG) 1, or not 0.

rr_last_update_time

Internal use. Not documented.

rr_last_update_days

The number of days since the record was last updated.

rr_name_id

Internal use. Not documented.

rr_value_id

Internal use. Not documented.

rr_type_id

Internal use. Not documented.

rr_glue_id

Internal use. Not documented.

rr_class_parameters

The class parameters applied to the resource record, encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

rr_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
rr_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

rr_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

dnsview_class_parameters

The class parameters applied to the DNS view the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dnsview_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
dnsview_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dnsview_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Example

In the example below, we call the service **dns_rr_list** with Python (Requests) and the clause WHERE to list only NS records.

Example 36.2. Calling the service dns_rr_list using Python

```
import requests

url = "https://solid.intranet/rest/dns_rr_list"

querystring = {"WHERE/rr_type": "NS"}

headers = {
    'x-ipm-username': "aXBtYWRtaW4=",
    'x-ipm-password': "YWRtaW4=",
    'cache-control': "no-cache"
}

response = requests.request("GET", url, headers=headers, params=querystring)

print(response.text)
```

Name

dns_rr_info — Display the properties of a resource record

Description

This service allows you to display the properties of a resource record.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Adding some resource records requires to specify one or more values:

Table 36.7. Expected values for the DNS records

| Type | Value number | Related field(s) | Description |
|-------|--------------|------------------|---|
| SOA | 1 | Name server | The FQDN of the primary Master name server for the zone the record belongs to. Has a special meaning when used with Dynamic DNS (DDNS): called MNAME, it allows the DNS client to know on which DNS server it has to update itself with DDNS. |
| | 2 | Responsible | The administrator email address for the zone the record belongs to. |
| | 3 | Serial number | The serial number for the zone the record belongs to. The serial number is automatically incremented for each zone change. |
| | 4 | Refresh | The refresh delay for the zone the record belongs to, in seconds. When reached, it forces the slave name server(s) to read the SOA record. If this record is higher than the slave's one, a zone transfer will be triggered by the slave to get the latest version of the zone. Typical values are 3 to 24 hours. |
| | 5 | Retry | The retry delay for the zone the record belongs to, in seconds. When reached, it forces the slave server to retry the request if it fails to reach the master server during a refresh cycle. Typical values are from 10 to 60 minutes. |
| | 6 | Expiration | The expiration time for the zone the record belongs to, in seconds. When reached, the zone records are considered to be no longer valid/authoritative. The DNS server then stops responding to queries for the zone. To avoid a major outage, the typical value is pretty high, between 1 to 3 weeks. |
| | 7 | Minimum | The minimum time for the zone the record belongs to, in seconds. It defines the period of time that negative responses can be cached from the slave. For instance, if a request cannot be resolved, the server will answer with a NXDOMAIN result (No such domain). The server will continue returning this value until the Minimum value expires, then it will retry the resolution. This value has to be between 0 and 3 hours. |
| NS | 1 | DNS server | The DNS server hostname. |
| MX | 1 | Preference | A number, between 0 and 65535, to define which server has priority if there are several RRs in the zone. The lowest the value has the priority over the other server(s). |
| | 2 | Mail server | The SMTP (mail) server hostname. |
| A | 1 | IP address | The IPv4 Address of the host. |
| AAAA | 1 | IPv6 address | The IPv6 Address of the host. |
| PTR | 1 | Localization | The hostname that should be returned when the IP address is queried. |
| CNAME | 1 | Hostname | The hostname. |
| TXT | 1 | Text | The description of your choice (max. 255 characters including spaces). |

| Type | Value number | Related field(s) | Description |
|--|--------------|-------------------|---|
| SRV | 1 | Priority | A number, between 0 and 65535, to define which server has priority if there are several SRV RRs in the zone. The lowest the value has the priority over the other server(s). |
| | 2 | Weight | A number, between 0 and 65535, that defines the server weight. If two SRV RRs have the same priority, the weight defines which server is more used. The field gives priority to the SRV RR with the greatest weight value: the greater the value is, the more the server is solicited. If you type in 0, there is no weighting. |
| | 3 | Ports | The port number that delivers the service to the target. |
| | 4 | Target | The hostname of the server delivering the service. |
| HINFO | 1 | CPU | The name of the CPU, either INTEL, AMD, SPARC, ALPHA, HPPA, POWERPC, MIPS, MOTOROLA or Other. |
| | 2 | OS | The name of the operating system, either AIX, FREEBSD, HPUX, IRIX, LINUX, OSF, OS/2, SOLARIS, SUNOS, VMS, WINDOWS, or Other. |
| MINFO | 1 | Responsible email | The email address of the administrator of the mail list. |
| | 2 | Error email | The email address that should receive the error messages regarding the mail list. |
| DNAME | 1 | Domain | The domain name of a subdomain of the zone. |
| AFSDB | 1 | Preference | Type the version of AFS service used: 1 (AFS version 3.0) or 2 (OSF DCE/NCA version). |
| | 2 | AFS server | The AFS hostname. |
| NAPTR | 1 | Order | A number, between 0 and 65535, to define which RR has priority if there are several NAPTR RRs in the zone. The lowest the value has the priority over the other record(s). |
| | 2 | Preference | A number, between 0 and 65535, to define which RR has priority if there are several NAPTR RRs have the same order in the zone. The lowest the value has the priority over the other record(s). |
| | 2 | Flags | The string that corresponds to the action you want your client application to perform. The flag specified impacts the data expected in the field Services, Regex and/or Replace. |
| | 3 | Services | The services parameters to which applies the action specified in the field Flags. You must respect your client application syntax. |
| | 4 | Regex | The string that contains a substitution expression matching the format <delimit ereg delimit substitution delimit flag> to which applies the action specified the field Flags. |
| | 5 | Replace | An FQDN domain name to which applies the action specified the field Flags. You can specify no domain name if you type in . (dot) in the field. |
| For more details, refer to RFC 3403 available on IETF website at http://tools.ietf.org/html/rfc3403 . | | | |
| NSAP | 1 | Name | The NSAP address of the end system. It should start with 0x and not exceed 255 hexadecimal characters separated by dots. |
| DS | 1 | Key Tag | The parent zone DS key tag. |
| | 2 | Key Algorithm | The parent zone DS algorithm key. |
| | 3 | Digest Type | The parent zone DS digest type. |
| | 4 | Digest | The parent zone DS digest. |
| DNSKEY | 1 | Flags | The zone key flag. |
| | 2 | Protocol | The protocol value. |
| | 3 | Algorithm | The public key's cryptographic algorithm. |
| | 4 | Key | The public key material. |

| Type | Value number | Related field(s) | Description |
|------|--------------|------------------|---|
| WKS | 1 | IP address | The IPv4 Address of the host that contains the services listed in the Services field. |
| | 2 | Protocol | The communication protocol, either <i>TCP</i> or <i>UDP</i> . |
| | 3 | Services | The list of needed services. |

Mandatory Input Parameters

rr_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...* .

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

rr_id

The database identifier (ID) of the DNS resource record, a unique numeric key value automatically incremented when you add a DNS RR. Use the ID to specify the DNS RR of your choice.

Output Parameters

rr_all_value

The concatenated values of the DNS resource record, as follows: *<value1>, <value2>, <value3>, <value4>, <value5>, <value6>, <value7>*.

dnszone_sort_zone

Internal use. Not documented.

dnszone_is_rpz

The RPZ status of the DNS zone the resource record belongs to. 1 indicates that the DNS zone the record belongs to is a Response Policy Zone.

dnszone_type

The type of the DNS zone the object belongs to, either *master*, *slave*, *forward*, *stub*, *hint* or *delegation-only*.

rr_full_name

The full name of the DNS resource record.

rr_full_name_utf

The name of the DNS resource record in UTF-8 format.

rr_name_ip_addr

Internal use. Not documented.

rr_name_ip4_addr

Internal use. Not documented.

rr_value_ip_addr

Internal use. Not documented.

rr_value_ip4_addr

Internal use. Not documented.

rr_glue

The shortname of the DNS resource record.

rr_type

The type of the DNS resource record.

Table 36.8. rr_type possible values

| Value | Record type description |
|--------|---|
| SOA | Start of Authority. Defines the zone name, an email contact and various time and refresh values applicable to the zone. It is automatically generated upon creation of a zone and cannot be added manually. |
| NS | Name Server. Defines the authoritative name server(s) for the domain (defined by the SOA record) or the subdomain. The NS record that indicates which server has authority over a zone is automatically generated upon the creation of a zone, once the server has been synchronized. |
| A | IPv4 Address. An IPv4 address for a host. |
| PTR | Pointer Record. Address Resolution, from an IP address (IPv4 or IPv6) to a host. Used in reverse mapping. |
| AAAA | IPv6 Address. An IPv6 address for a host. |
| CNAME | Canonical Name. An alias name for a host. |
| MX | Mail Exchange. The mail server/exchanger that services this zone. |
| SRV | Services record. Defines services available in the zone, for example, LDAP, HTTP, etc... |
| DNAME | Delegation of Reverse Names. Delegation of reverse addresses primarily in IPv6. (Deprecated, use the CNAME RR instead) |
| TXT | Text. Information associated with a name. |
| DS | Delegation Signer, a DNSSEC related RR used to verify the validity of the ZSK of a subdomain. |
| DNSKEY | DNS Key. It contains the public cryptographic key used to sign the zone with DNSSEC. |
| 65534 | A private type record automatically added to the zone once it is signed with DNSSEC. |
| HINFO | System Information. Information about a host: hardware type and operating system description. |
| MINFO | Mailbox mail list Information. Defines the mail administrator for a mail list and optionally a mailbox to receive error messages relating to the mail list. |
| AFSDB | AFS Database. Location of the AFS servers. |
| WKS | Well-Known Service. Defines the services and protocols supported by a host. (Deprecated, use the SRV RR instead) |
| NAPTR | Naming Authority Pointer Record. General purpose definition of rule set to be used by applications e.g. VoIP. |
| NSAP | Network Service Access Point. Defines record (equivalent of an A record) maps a host name to an endpoint address. |

ttl

The time to live of the DNS resource record, in seconds.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

rr_class_name

The name of the class applied to the resource record, it can be preceded by the class directory.

value1

The first or only value required for the DNS resource record, as detailed in the service description.

value2

The second value of the DNS resource record, depending on its type, as detailed in the service description.

value3

The third value of the DNS resource record, depending on its type, as detailed in the service description.

value4

The fourth value of the DNS resource record, depending on its type, as detailed in the service description.

value5

The fifth value of the DNS resource record, depending on its type, as detailed in the service description.

value6

The sixth value of the DNS resource record, depending on its type, as detailed in the service description.

value7

The seventh value of the DNS resource record, depending on its type, as detailed in the service description.

dnszone_id

The database identifier (ID) of the DNS zone the object belongs to.

rr_id

The database identifier (ID) of the DNS resource record.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

dnszone_name_utf

The name of the DNS zone the resource record belongs to, in UTF-8 format.

dnszone_name

The name of the DNS zone the object belongs to.

dns_name

The name of the DNS server the object belongs to.

dns_type

The type of the DNS server the object belongs to.

Table 36.9. *dns_type* possible values

| Type | Description |
|----------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msdaemon | Microsoft Windows DNS server |

| Type | Description |
|-------|------------------------------------|
| aws | Amazon Route 53 server |
| other | Generic DNS server |
| vdns | EfficientIP DNS smart architecture |

dns_cloud

Internal use. Not documented.

vdns_parent_id

The database identifier (ID) of the DNS smart architecture managing the DNS server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dnsview_name

The name of the DNS view the object belongs to.

dnsview_class_name

The name of the class applied to the DNS view the object belongs to, it can be preceded by the class directory.

dnsview_id

The database identifier (ID) of the DNS view the object belongs to.

dnszone_site_name

The name of the space associated with the DNS zone the RR belongs to.

dnszone_is_reverse

A way to determine if the DNS zone the resource record belongs to provides reverse resolution (1) or direct/name resolution (0),

dnszone_masters

For resource records in slave DNS zones, the IP address of the DNS server and, if relevant, the name of the DNS view that contain the master DNS zone, as follows: <ip_addr>; or <ip_addr> key <dnsview_name>; .

vdns_parent_name

The name of the DNS smart architecture managing the DNS server the object belongs to. # indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dnszone_forwarders

The IP address(es) of the forwarder(s) associated with the DNS zone the resource record belongs to. It lists the DNS servers to which any unknown query on this zone should be sent, as follows: <ip_address1>;<ip_address2>;... .

dns_class_name

The name of the class applied to the DNS server the object belongs to, it can be preceded by the class directory.

dnszone_class_name

The name of the class applied to the DNS zone the object belongs to, it can be preceded by the class directory.

dns_version

The version details of the DNS server the object belongs to.

dns_comment

The description of the DNS server the object belongs to.

delayed_create_time

The delay of creation status. 1 indicates that the object is not created yet.

delayed_delete_time

The delay of deletion status. 1 indicates that the object is not deleted yet.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 36.10. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

rr_auth_gsstsig

A way to determine if the record was authenticated via dynamic update (GSS-TSIG) 1, or not 0.

rr_last_update_time

Internal use. Not documented.

rr_last_update_days

The number of days since the record was last updated.

rr_name_id

Internal use. Not documented.

rr_value_id

Internal use. Not documented.

rr_type_id

Internal use. Not documented.

rr_glue_id

Internal use. Not documented.

rr_class_parameters

The class parameters applied to the resource record, encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

rr_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **rr_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

rr_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

dnsview_class_parameters

The class parameters applied to the DNS view the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

dnsview_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **dnsview_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

dnsview_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Example

In the example below, we call the service **dns_rr_info** with Ruby (NET::Http) to display the properties a resource record.

Example 36.3. Calling the service dns_rr_info using Ruby

```
require 'uri'
require 'net/http'

url = URI("https://solid.intranet/rest/dns_rr_info?rr_id=204")

http = Net::HTTP.new(url.host, url.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE

request = Net::HTTP::Get.new(url)
request["x-ipm-username"] = 'aXBtYWRtaW4='
request["x-ipm-password"] = 'YWRtaW4='
request["cache-control"] = 'no-cache'

response = http.request(request)
puts response.read_body
```

Name

dns_rr_count — Count the number of resource records

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dns_rr_groupby — Group resource records by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

dns_rr_groupby_count — Count the number of resource records grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

dns_rr_delete — Delete a resource record

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(rr_id || (rr_name && (dns_id || dns_name || hostaddr)))

Input Parameters

rr_id

The database identifier (ID) of the DNS resource record, a unique numeric key value automatically incremented when you add a DNS RR. Use the ID to specify the DNS RR of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnszone_id

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

zone

Deprecated, replaced by **dnszone_name**.

dnszone_name

The name of the DNS zone the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_id

The database identifier (ID) of the DNS view, a unique numeric key value automatically incremented when you add a DNS view. Use the ID to specify the DNS view of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_name

The name of the DNS view.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_name

The name of the DNS resource record.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_ttl

Deprecated, replaced by **ttl**.

rr_type

The type of the DNS resource record.

Table 36.11. rr_type possible values

| Value | Record type description |
|-------|---|
| SOA | Start of Authority. Defines the zone name, an email contact and various time and refresh values applicable to the zone. It is automatically generated upon creation of a zone and cannot be added manually. |
| NS | Name Server. Defines the authoritative name server(s) for the domain (defined by the SOA record) or the subdomain. The NS record that indicates which server has authority over a zone is automatically generated upon the creation of a zone, once the server has been synchronized. |
| A | IPv4 Address. An IPv4 address for a host. |
| PTR | Pointer Record. Address Resolution, from an IP address (IPv4 or IPv6) to a host. Used in reverse mapping. |
| AAAA | IPv6 Address. An IPv6 address for a host. |
| CNAME | Canonical Name. An alias name for a host. |
| MX | Mail Exchange. The mail server/exchanger that services this zone. |
| SRV | Services record. Defines services available in the zone, for example, LDAP, HTTP, etc... |
| DNAME | Delegation of Reverse Names. Delegation of reverse addresses primarily in IPv6. (Deprecated, use the CNAME RR instead) |

| Value | Record type description |
|--------|---|
| TXT | Text. Information associated with a name. |
| DS | Delegation Signer, a DNSSEC related RR used to verify the validity of the ZSK of a subdomain. |
| DNSKEY | DNS Key. It contains the public cryptographic key used to sign the zone with DNSSEC. |
| 65534 | A private type record automatically added to the zone once it is signed with DNSSEC. |
| HINFO | System Information. Information about a host: hardware type and operating system description. |
| MINFO | Mailbox mail list Information. Defines the mail administrator for a mail list and optionally a mailbox to receive error messages relating to the mail list. |
| AFSDB | AFS Database. Location of the AFS servers. |
| WKS | Well-Known Service. Defines the services and protocols supported by a host. (Deprecated, use the SRV RR instead) |
| NAPTR | Naming Authority Pointer Record. General purpose definition of rule set to be used by applications e.g. VoIP. |
| NSAP | Network Service Access Point. Defines record (equivalent of an A record) maps a host name to an endpoint address. |

Note that the parameter is not case sensitive, you could type in A or a.

| | | | |
|---------------|---|----------------|-----|
| Type | Fixed value: A NS MD MF CNAME SOA MB MG MR NULL WKS PTR HINFO MINFO MX TXT SPF RP AFSDB X25 ISDN RT NSAP NSAP_PTR SIG KEY PX GPOS AAAA LOC NXT EID NIMLOC SRV ATMA NAPTR KX CERT A6 DNAME OPT DS DNSSIG NSEC DNSKEY NSEC3 NSEC3PARAM CDS CDNSKEY CAA TLSA SSHFP OPENPGPKEY URI AVC NINFO DLV DHCID EUI48 EUI64 NID L32 L64 HTTPS SVCB | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value1

The first or only value required for the DNS resource record, as detailed in the service description

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value2

The second value of the DNS resource record, depending on its type, as detailed in the service description

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value3

The third value of the DNS resource record, depending on its type, as detailed in the service description

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

rr_value4

The fourth value of the DNS resource record, depending on its type, as detailed in the service description

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value5

The fifth value of the DNS resource record, depending on its type, as detailed in the service description

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value6

The sixth value of the DNS resource record, depending on its type, as detailed in the service description

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rr_value7

The seventh value of the DNS resource record, depending on its type, as detailed in the service description

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_site_id

The database identifier (ID) of the space associated with the DNS zone the record belongs to.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Example

In the example below, we call the service **dns_rr_delete** with PHP (cURL) to delete a record from the database.

Example 36.4. Calling the service dns_rr_delete using PHP

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/dns_rr_delete?rr_id=247",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "DELETE",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Chapter 37. DNS ACL

Name

dns_acl_add — Add/Edit a DNS ACL

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** ((dnsacl_name && (dns_id || dns_name || hostaddr)) && dnsacl_value)
- **Editing:** ((dnsacl_id || (dnsacl_name && (dns_id || dns_name || hostaddr))) && dnsacl_value)

Input Parameters

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_addr

Deprecated, replaced by **hostaddr**.

dnsacl_name

The name of the DNS ACL, each DNS ACL must have a unique name.

| Type | String | Maximum length | 64 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

acl_name

Deprecated, replaced by **dnsacl_name**.

dnsacl_value

The values of the DNS ACL in order of priority, as follows: <value_1>;<value_2>....

| | | | |
|---------------|--------------------------------|----------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

acl_value

Deprecated, replaced by **dnsacl_value**.

dnsacl_id

The database identifier (ID) of the DNS ACL, a unique numeric key value automatically incremented when you add a DNS ACL. Use the ID to specify the DNS ACL of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dns_acl_list — List the DNS ACLs

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dnsacl_name

The name of the DNS ACL.

dnsacl_value

The values of the DNS ACL in order of priority, as follows: <value_1>;<value_2>... .

dnsacl_id

The database identifier (ID) of the DNS ACL.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

Name

dns_acl_info — Display the properties of a DNS ACL

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

`dnsacl_id`

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

`dnsacl_id`

The database identifier (ID) of the DNS ACL, a unique numeric key value automatically incremented when you add a DNS ACL. Use the ID to specify the DNS ACL of your choice.

Output Parameters

`dnsacl_name`

The name of the DNS ACL.

`dnsacl_value`

The values of the DNS ACL in order of priority, as follows: `<value_1>;<value_2>....`

`dnsacl_id`

The database identifier (ID) of the DNS ACL.

`dns_id`

The database identifier (ID) of the DNS server the object belongs to.

Name

dns_acl_count — Count the number of DNS ACLs

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dns_acl_delete — Delete a DNS ACL

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dnsacl_id

Input Parameters

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnsacl_id

The database identifier (ID) of the DNS ACL, a unique numeric key value automatically incremented when you add a DNS ACL. Use the ID to specify the DNS ACL of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnsacl_name

The name of the DNS ACL.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 38. TSIG Key

Name

dns_key_add — Add/Edit a TSIG key

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (dnskey_name && (dns_id || dns_name || hostaddr))
- **Editing:** (dnskey_name && (dns_id || dns_name || hostaddr))

Input Parameters

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_addr

Deprecated, replaced by **hostaddr**.

dnskey_name

The name of the DNS TSIG key, each DNS TSIG key must have a unique name.

| Type | String | Maximum length | 63 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnskey_proto

The encryption protocol of the TSIG key.

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: hmac-md5 hmac-sha1 hmac-sha224 hmac-sha256 hmac-sha384 hmac-sha512 | Maximum length | N/A |
| Default value | hmac-md5 | Can be edited | Yes |

dnskey_value

The value of the TSIG key.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_name

The name of the DNS view.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnsview_id

The database identifier (ID) of the DNS view, a unique numeric key value automatically incremented when you add a DNS view. Use the ID to specify the DNS view of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

dnskey_id

The database identifier (ID) of the DNS TSIG key, a unique numeric key value automatically incremented when you add a DNS TSIG key. Use the ID to specify the DNS TSIG key of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

dns_key_list — List the DNS keys

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dns_id

The database identifier (ID) of the DNS server the object belongs to.

dns_name

The name of the DNS server the object belongs to.

dns_type

The type of the DNS server the object belongs to.

Table 38.1. *dns_type* possible values

| Type | Description |
|----------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msdaemon | Microsoft Windows DNS server |
| aws | Amazon Route 53 server |
| other | Generic DNS server |
| vdns | EfficientIP DNS smart architecture |

vdns_parent_id

The database identifier (ID) of the DNS smart architecture managing the DNS server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dnskey_id

The database identifier (ID) of the DNS TSIG key.

dnskey_name

The name of the DNS TSIG key.

dnskey_value

The value of the TSIG key.

dnskey_proto

The encryption protocol of the TSIG key.

dnsview_name

The name of the DNS view associated with the DNS TSIG key.

dnsview_id

The database identifier (ID) of the DNS view associated with the DNS TSIG key.

Name

dns_key_info — Display the properties of a TSIG key

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

dnskey_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

dnskey_id

The database identifier (ID) of the DNS TSIG key, a unique numeric key value automatically incremented when you add a DNS TSIG key. Use the ID to specify the DNS TSIG key of your choice.

Output Parameters

dns_id

The database identifier (ID) of the DNS server the object belongs to.

dns_name

The name of the DNS server the object belongs to.

dns_type

The type of the DNS server the object belongs to.

Table 38.2. dns_type possible values

| Type | Description |
|----------|---|
| ipm | EfficientIP or EfficientIP Package server |
| msdaemon | Microsoft Windows DNS server |
| aws | Amazon Route 53 server |
| other | Generic DNS server |

| Type | Description |
|------|------------------------------------|
| vdns | EfficientIP DNS smart architecture |

vdns_parent_id

The database identifier (ID) of the DNS smart architecture managing the DNS server the object belongs to. 0 indicates that the server the object belongs to is not managed by a smart architecture or is a smart architecture itself.

dnskey_id

The database identifier (ID) of the DNS TSIG key.

dnskey_name

The name of the DNS TSIG key.

dnskey_value

The value of the TSIG key.

dnskey_proto

The encryption protocol of the TSIG key.

dnsview_name

The name of the DNS view associated with the DNS TSIG key.

dnsview_id

The database identifier (ID) of the DNS view associated with the DNS TSIG key.

Name

dns_key_count — Count the number of TSIG keys

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

dns_key_delete — Delete a TSIG key

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dnskey_id || (dnskey_name && (dns_id || dns_name || hostaddr)))

Input Parameters

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the DNS server.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnskey_id

The database identifier (ID) of the DNS TSIG key, a unique numeric key value automatically incremented when you add a DNS TSIG key. Use the ID to specify the DNS TSIG key of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dnskey_name

The name of the DNS TSIG key.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 39. DNSSEC

Name

dnssec_zone_keys_list — List the DNSSEC keys of a zone

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

dns_name

The name of the DNS server the object belongs to.

dns_id

The database identifier (ID) of the DNS server the object belongs to.

dnszone_id

The database identifier (ID) of the DNS zone the object belongs to.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

obj_type

The type of the DNSSEC key, either *zsk* or *ksk* or *anchor*.

encryption_data

The private part of the DNSSEC key data.

encryption_bits

The number of bits over which the DNSSEC key is encrypted.

validity

The value of the DNSSEC key validity period, in days.

start_date

The time at which the DNSSEC key starts being valid, in decimal UNIX date format.

encryption_data_public

The public part of the DNSSEC key data.

obj_name

The name of the DNSSEC key, specified as follows : <a>.+<c>+<d> , where:

- a. is either *Z* for ZSK or *K* for KSK keys.
 - b. is the <*dnszone_name*>.
 - c. is the key algorithm.
 - d. is the key tag.
- a can be either *Z* for ZSK or *K* for KSK keys.
 - b is the <*dnszone_name*>.
 - c is the key algorithm.
 - d is the key tag.

ds

The delegation signer(s) (DS) associated with the zone, as follows: [<dnszone_name>. IN DS <key_algorithm> <key_tag> <algorithm_key> <digest_type> <digest>] [<dnszone_name>. IN DS <key_algorithm> <key_tag> <algorithm_key> <digest_type> <digest>] ...

dlv

Internal use. Not documented.

module

The name of the signature module. By default, it is set to *dnssec*.

dnszone_name

The name of the DNS zone the object belongs to.

encryption_type

The type of encryption to used for the DNSSEC key, either *rsasha256*, *rsasha512*, *nsec3rsasha1* or *nsec3dsa*.

Name

dnssec_zone_keys_info — Display the properties of a DNSSEC ZSK, KSK or Trust Anchor

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

`dnszone_id`

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

`dnszone_id`

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

Output Parameters

`dns_name`

The name of the DNS server the object belongs to.

`dns_id`

The database identifier (ID) of the DNS server the object belongs to.

`dnszone_id`

The database identifier (ID) of the DNS zone the object belongs to.

`row_enabled`

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, `row_enabled` is set to 1 when an object is created.

obj_type

The type of the DNSSEC key, either *zsk* or *ksk* or *anchor*.

encryption_data

The private part of the DNSSEC key data.

encryption_bits

The number of bits over which the DNSSEC key is encrypted.

validity

The value of the DNSSEC key validity period, in days.

start_date

The time at which the DNSSEC key starts being valid, in decimal UNIX date format.

encryption_data_public

The public part of the DNSSEC key data.

obj_name

The name of the DNSSEC key, specified as follows : <a>.+<c>+<d> , where:

- a. is either *Z* for ZSK or *K* for KSK keys.
- b. is the <*dnszone_name*>.
- c. is the key algorithm.
- d. is the key tag.
 - a can be either *Z* for ZSK or *K* for KSK keys.
 - b is the <*dnszone_name*>.
 - c is the key algorithm.
 - d is the key tag.

ds

The delegation signer(s) (DS) associated with the zone, as follows: [<*dnszone_name*>. *IN DS* <*key_algorithm*> <*key_tag*> <*algorithm_key*> <*digest_type*> <*digest*>] [<*dnszone_name*>. *IN DS* <*key_algorithm*> <*key_tag*> <*algorithm_key*> <*digest_type*> <*digest*>] ...

dlv

Internal use. Not documented.

module

The name of the signature module. By default, it is set to *dnssec*.

dnszone_name

The name of the DNS zone the object belongs to.

encryption_type

The type of encryption to used for the DNSSEC key, either *rsasha256*, *rsasha512*, *nsec3rsasha1* or *nsec3dsa*.

Name

dnssec_enable_sign_zone — Sign a zone with DNSSEC

Description

This service allows you to specify a zone and sign it with DNSSEC, only zones belonging to a smart architecture or an EfficientIP DNS server can be signed. Once a zone is signed, the server it belongs to becomes authoritative and every transaction with the zone must be handled via DNSSEC.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(dnszone_id || (dnszone_name && (dns_id || dns_name || dns_hostaddr)))

Input Parameters

zsk_keyring_ids

Internal use. Not documented.

| Type | List of strings separated by ; | Maximum length | N/A |
|---------------|--------------------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ksk_keyring_ids

Internal use. Not documented.

| Type | List of strings separated by ; | Maximum length | N/A |
|---------------|--------------------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

zsk_encryption_type

The type of encryption to use for the ZSK public and private keys generation, either *rsasha256*, *rsasha512*, *nsec3rsasha1* or *nsec3dsa*

| Type | Fixed value: ecdsap256sha256 ecdsap384sha384 ed25519 ed448 rsasha256 rsasha512 nsec3rsasha1 nsec3dsa | Maximum length | N/A |
|---------------|---|----------------|-----|
| Default value | N/A | Can be edited | Yes |

zsk_encryption_bits

The number of bits used to generate the ZSK:

- For RSASHA encryption, set it between 512 and 4096 bits. The value should be a multiple of 64.
- For DSA encryption, you can set from 512 to 1024 bits. The value should be a multiple of 64.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

zsk_validity

The value of the ZSK validity period. Use the parameter *zsk_validity_unit* to indicate the corresponding time unit.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

zsk_validity_unit

The time unit of the ZSK validity period, either *day*, *month*, *year* or, with no need to indicate the *zsk_validity* value, *infinity*.

| | | | |
|----------------------|---|-----------------------|-----|
| Type | Fixed value: day month year infinity | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ksk_encryption_type

The type of encryption to use for the KSK public and private keys generation, either *rsasha256*, *rsasha512*, *nsec3rsasha1* or *nsec3dsa*

| | | | |
|----------------------|---|-----------------------|-----|
| Type | Fixed value: ecdsap256sha256 ecdsap384sha384 ed25519 ed448 rsasha256 rsasha512 nsec3rsasha1 nsec3dsa | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ksk_encryption_bits

The number of bits over which the KSK keys should be encrypted:

- For RSASHA*, you can set from 512 to 4096 bits.
- For DSA, you can set from 512 to 1024 bits and a modulus of 64.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ksk_validity

The value of the KSK validity period. Use the parameter *ksk_validity_unit* to indicate the corresponding time unit.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ksk_validity_unit

The time unit of the KSK validity period, either *day*, *month*, *year* or, with no need to indicate the *ksk_validity* value, *infinity*.

| | | | |
|----------------------|---|-----------------------|-----|
| Type | Fixed value: day month year infinity | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

alert_snmp_params_oid

Internal use. Not documented.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

alert_released_snmp_trap_oid

Internal use. Not documented.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

alert_snmp_params_community

Internal use. Not documented.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

alert_snmp_params_dest

Internal use. Not documented.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

alert_snmp_params_version

Internal use. Not documented.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

alert_trap_snmp

Internal use. Not documented.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

alert_additional_mail

Internal use. Not documented.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

alert_send_mail

Internal use. Not documented.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

alert_group_mail

Internal use. Not documented.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_name

The name of the DNS zone the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the DNS server.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the DNS server, a unique numeric key value automatically incremented when you add a DNS server. Use the ID to specify the DNS server of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_hostaddr

The IP address of the DNS server the zone you want to sign belongs to.

| | | | |
|---------------|-------------------|----------------|-----|
| Type | IPv4/Ipv6 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dnszone_id

The database identifier (ID) of the DNS zone, a unique numeric key value automatically incremented when you add a DNS zone. Use the ID to specify the DNS zone of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

source_mail_registry

Internal use. Not documented.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

source_mail_addr

Internal use. Not documented.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

param_value

The value of the input parameter(s) that caused the error during the service execution.

param_name

Internal use. Not documented.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

Part V. Network Object Manager Services

Table of Contents

| | |
|---|-----|
| 40. Network Object Manager Folder | 747 |
| nom_folder_add | 748 |
| nom_folder_list | 752 |
| nom_folder_info | 755 |
| nom_folder_count | 757 |
| nom_folder_delete | 758 |
| 41. Network Object | 760 |
| nom_netobj_add | 761 |
| nom_netobj_list | 765 |
| nom_netobj_info | 768 |
| nom_netobj_count | 771 |
| nom_netobj_connected_list | 772 |
| nom_netobj_connected_count | 774 |
| nom_netobj_delete | 775 |
| 42. Interface | 777 |
| nom_iface_add | 778 |
| nom_iface_list | 782 |
| nom_iface_info | 786 |
| nom_iface_count | 789 |
| nom_port_list | 790 |
| nom_port_info | 792 |
| nom_port_count | 793 |
| nom_iface_delete | 794 |

Chapter 40. Network Object Manager Folder

Name

nom_folder_add — Add/Edit a Network Object Manager folder

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (nomfolder_path || nomfolder_name)
- **Editing:** (nomfolder_id || nomfolder_path || nomfolder_name)

Input Parameters

nomfolder_id

The database identifier (ID) of the Network Object Manager folder, a unique numeric key value automatically incremented when you add a folder. Use the ID to specify the folder of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomfolder_name

The name of the Network Object Manager folder. Use the name to specify which folder to edit.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

nomfolder_path

The path to the Network Object Manager folder. Use the path to specify which folder to edit.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomfolder_description

The description of the Network Object Manager folder.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

nomfolder_site_name

The name of an existing space you want to associate with the Network Object Manager folder.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | # | Can be edited | Yes |

parent_nomfolder_id

The database identifier (ID) of the parent folder of the Network Object Manager folder you are adding, a unique numeric key value automatically incremented when you add a folder. You can only specify it when adding a folder.

| | | | |
|---------------|---------|----------------|-----|
| Type | Integer | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

parent_nomfolder_path

The path to the parent folder of the Network Object Manager folder.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomfolder_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

nomfolder_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomfolder_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: **<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....**. If the inheritance or propagation property is not specified, its default value - *set, propagate* - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

nom_folder_list — List the Network Object Manager folders

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

nomfolder_id

The database identifier (ID) of the Network Object Manager folder.

parent_nomfolder_id

The database identifier (ID) of the parent folder of the Network Object Manager folder.

nomfolder_name

The name of the Network Object Manager folder.

nomfolder_level

The level of the Network Object Manager folder, where 0 represents the highest level in the folder hierarchy.

nomfolder_path

The path to the Network Object Manager folder.

nomfolder_description

The description of the Network Object Manager folder.

nomfolder_site_name

The name of the space associated with the Network Object Manager folder.

nomfolder_class_name

The name of the class applied to the object, it can be preceded by the class directory.

nomfolder_nb_netobj

The total number of network objects of the Network Object Manager folder.

parent_nomfolder_name

The name of the parent folder of the Network Object Manager folder.

parent_nomfolder_path

The path to the parent folder of the Network Object Manager folder.

parent_nomfolder_class_name

The name of the class applied to the parent folder of the Network Object Manager folder, it can be preceded by the class directory.

nomfolder_class_parameters

The class parameters applied to the Network Object Manager folder and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

nomfolder_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **nomfolder_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

nomfolder_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

parent_nomfolder_class_parameters

The class parameters applied to the parent folder of the Network Object Manager folder and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

parent_nomfolder_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **parent_nomfolder_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

nom_folder_info — Display the properties of a Network Object Manager folder

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

nomfolder_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

nomfolder_id

The database identifier (ID) of the Network Object Manager folder, a unique numeric key value automatically incremented when you add a folder. Use the ID to specify the folder of your choice.

Output Parameters

nomfolder_id

The database identifier (ID) of the Network Object Manager folder.

parent_nomfolder_id

The database identifier (ID) of the parent folder of the Network Object Manager folder.

nomfolder_name

The name of the Network Object Manager folder.

nomfolder_level

The level of the Network Object Manager folder, where 0 represents the highest level in the folder hierarchy.

nomfolder_path

The path to the Network Object Manager folder.

nomfolder_description

The description of the Network Object Manager folder.

nomfolder_site_name

The name of the space associated with the Network Object Manager folder.

nomfolder_class_name

The name of the class applied to the object, it can be preceded by the class directory.

nomfolder_nb_netobj

The total number of network objects of the Network Object Manager folder.

parent_nomfolder_name

The name of the parent folder of the Network Object Manager folder.

parent_nomfolder_path

The path to the parent folder of the Network Object Manager folder.

parent_nomfolder_class_name

The name of the class applied to the parent folder of the Network Object Manager folder, it can be preceded by the class directory.

nomfolder_class_parameters

The class parameters applied to the Network Object Manager folder and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

nomfolder_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **nomfolder_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

nomfolder_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

parent_nomfolder_class_parameters

The class parameters applied to the parent folder of the Network Object Manager folder and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

parent_nomfolder_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **parent_nomfolder_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

nom_folder_count — Count the number of Network Object Manager folders

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

nom_folder_delete — Delete a Network Object Manager folder

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

nomfolder_id || nomfolder_path || nomfolder_name

Input Parameters

nomfolder_id

The database identifier (ID) of the Network Object Manager folder, a unique numeric key value automatically incremented when you add a folder. Use the ID to specify the folder of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomfolder_name

The name of the Network Object Manager folder. Use the name to specify the folder of your choice.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomfolder_path

The path to the Network Object Manager folder. Use the path to specify the folder of your choice.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

parent_nomfolder_id

The database identifier (ID) of the parent folder of the Network Object Manager folder, a unique numeric key value automatically incremented when you add a folder.

| Type | Integer | Maximum length | N/A |
|---------------|---------|----------------|-----|
| Default value | 0 | Can be edited | Yes |

parent_nomfolder_path

The path to the parent folder of the Network Object Manager folder.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 41. Network Object

Name

nom_netobj_add — Add/Edit a network object

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (nomnetobj_name && nomfolder_id) || (nomnetobj_name && nomfolder_path)
- **Editing:** nomnetobj_id || (nomnetobj_name && nomfolder_id) || (nomnetobj_name && nomfolder_path)

Input Parameters

nomnetobj_id

The database identifier (ID) of the network object, a unique numeric key value automatically incremented when you add a network object. Use the ID to specify the network object of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomnetobj_name

The name of the network object. Use the name to specify which network object to edit.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

nomnetobj_description

The description of the network object.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

nomnetobj_type

The type of the network object. It cannot exceed 16 characters. If this type has not been specified yet, it is added.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 16 |
| Default value | | Can be edited | Yes |

nomnetobj_state

The state of the network object. It cannot exceed 16 characters. If this state has not been specified yet, it is added.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 16 |
| Default value | | Can be edited | Yes |

nomfolder_id

The database identifier (ID) of the Network Object Manager folder, a unique numeric key value automatically incremented when you add a folder. Use the ID to specify the folder of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomfolder_path

The path to the Network Object Manager folder. Use the path to specify the folder of your choice.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

parent_nomnetobj_id

The database identifier (ID) of the parent network object.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

parent_nomnetobj_name

The name of the parent network object.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

parent_nomfolder_path

The path to the parent folder of the Network Object Manager folder.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomnetobj_class_name

The name of the class to apply to the network object you are adding or editing. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

nomnetobj_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: *<class-parameter1>=<value1>&<class-parameter2>=<value2>&... .*

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomnetobj_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - *set, propagate* - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

nom_netobj_list — List the network objects

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

nomnetobj_id

The database identifier (ID) of the network object.

nomfolder_id

The database identifier (ID) of the Network Object Manager folder the object belongs to.

parent_nomnetobj_id

The database identifier (ID) of the parent network object of the network object.

nomnetobj_name

The name of the network object.

nomnetobj_path

The path to the network object.

nomnetobj_type

The type of the network object.

nomnetobj_state

The state of the network object.

nomnetobj_description

The description of the network object.

nomnetobj_class_name

The name of the class applied to the network object, it can be preceded by the class directory.

nomnetobj_nb_iface

The number of interfaces that belong to the network object.

nomnetobj_nb_connected_ports

The number of interface ports connected to the network object.

nomnetobj_main_iface

The IPv4 address of the Main interface of the network object.

nomnetobj_main_iface_hostaddr

The human readable version of the parameter **nomnetobj_main_iface**.

nomnetobj_main_iface6

The IPv6 address of the Main interface of the network object.

nomnetobj_main_iface6_hostaddr

The human readable version of the parameter **nomnetobj_main_iface6**.

nomnetobj_connected_nom

The name of the network object connected to the network object.

nomfolder_path

The path to the Network Object Manager folder the object belongs to.

nomfolder_name

The name of the Network Object Manager folder the object belongs to.

parent_nomnetobj_name

The name of the parent network object of the network object.

parent_nomfolder_path

The path to the parent folder of the Network Object Manager folder the object belongs to.

nomnetobj_class_parameters

The class parameters applied to the network object and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

nomnetobj_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **nomnetobj_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

nomnetobj_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

nomfolder_class_parameters

The class parameters applied to the Network Object Manager folder the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

nomfolder_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **nomfolder_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

nom_netobj_info — Display the properties of a network object

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

nomnetobj_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

nomnetobj_id

The database identifier (ID) of the network object, a unique numeric key value automatically incremented when you add a network object. Use the ID to specify the network object of your choice.

Output Parameters

nomnetobj_id

The database identifier (ID) of the network object.

nomfolder_id

The database identifier (ID) of the Network Object Manager folder the object belongs to.

parent_nomnetobj_id

The database identifier (ID) of the parent network object of the network object.

nomnetobj_name

The name of the network object.

nomnetobj_path

The path to the network object.

nomnetobj_type

The type of the network object.

nomnetobj_state

The state of the network object.

nomnetobj_description

The description of the network object.

nomnetobj_class_name

The name of the class applied to the network object, it can be preceded by the class directory.

nomnetobj_nb_iface

The number of interfaces that belong to the network object.

nomnetobj_nb_connected_ports

The number of interface ports connected to the network object.

nomnetobj_main_iface

The IPv4 address of the Main interface of the network object.

nomnetobj_main_iface_hostaddr

The human readable version of the parameter **nomnetobj_main_iface**.

nomnetobj_main_iface6

The IPv6 address of the Main interface of the network object.

nomnetobj_main_iface6_hostaddr

The human readable version of the parameter **nomnetobj_main_iface6**.

nomnetobj_connected_nom

The name of the network object connected to the network object.

nomfolder_path

The path to the Network Object Manager folder the object belongs to.

nomfolder_name

The name of the Network Object Manager folder the object belongs to.

parent_nomnetobj_name

The name of the parent network object of the network object.

parent_nomfolder_path

The path to the parent folder of the Network Object Manager folder the object belongs to.

nomnetobj_class_parameters

The class parameters applied to the network object and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

nomnetobj_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **nomnetobj_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

nomnetobj_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

nomfolder_class_parameters

The class parameters applied to the Network Object Manager folder the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

nomfolder_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **nomfolder_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

Name

nom_netobj_count — Count the number of network objects

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

nom_netobj_connected_list — List the network objects connected to a specific network object

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

nomnetobj_id

The database identifier (ID) of the connected network object, a unique numeric key value automatically incremented when you add a network object. Use the ID to specify the network object of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

nomfolder_id

The database identifier (ID) of the Network Object Manager folder the object belongs to.

nomfolder_path

The path to the Network Object Manager folder the object belongs to.

nomnetobj_id

The database identifier (ID) of the network object connected to the network object.

nomnetobj_name

The name of the network object.

nomnetobj_type

The type of the network object.

nomnetobj_main_iface_hostaddr

The IPv4 address of the Main interface of the network object, in human readable format.

nomnetobj_main_iface6_hostaddr

The IPv6 address of the Main interface of the network object, in human readable format.

nomnetobj_path

The path to the network object.

port_ids

The database identifier (ID) of the interface port of the network object connected to the network object.

port_names

The name of the interface port of the network object connected to the network object.

Name

nom_netobj_connected_count—Count the number of network objects connected to a specific network object

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

nomnetobj_id

The database identifier (ID) of the connected network object, a unique numeric key value automatically incremented when you add a network object. Use the ID to specify the network object of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

Name

nom_netobj_delete — Delete a network object

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

nomnetobj_id || (nomnetobj_name && nomfolder_id) || (nomnetobj_name && nomfolder_path)

Input Parameters

nomnetobj_id

The database identifier (ID) of the network object, a unique numeric key value automatically incremented when you add a network object. Use the ID to specify the network object of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomnetobj_name

The name of the network object. Use the name to specify the network object of your choice.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomfolder_id

The database identifier (ID) of the Network Object Manager folder, a unique numeric key value automatically incremented when you add a folder. Use the ID to specify the folder of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomfolder_path

The path to the Network Object Manager folder. Use the path to specify the folder of your choice.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 42. Interface

Name

nom_iface_add — Add/Edit an interface or an interface port

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (nomiface_port_name && (nomnetobj_id || (nomfolder_path && nomnetobj_name)))
- **Editing:** nomiface_id || (nomiface_port_name && (nomnetobj_id || (nomfolder_path && nomnetobj_name)))

Input Parameters

nomiface_id

The database identifier (ID) of the interface or interface port, a unique numeric key value automatically incremented when you add an interface or interface port. Use the ID to specify the interface or interface port of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomiface_name

The name of the interface or interface port.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomiface_hostaddr

The IPv4 or IPv6 address of the interface or interface port.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomiface_main

A way to set the interface as Main interface (1) or not (0). A network object can only have one Main IPv4 interface and one Main IPv6 interface at the most.

| Type | Boolean: 0 1 | Maximum length | N/A |
|---------------|-----------------|----------------|-----|
| Default value | 0 | Can be edited | Yes |

nomiface_port_name

The name of the interface port, as follows: *port_<number>*. Use the name to specify which interface or interface port to edit.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

nomiface_port_mac

The MAC address of the interface port. Use the MAC address to specify which interface or interface port to edit. This parameter is overwritten if you specify a **nomiface_mac**.

| | | | |
|---------------|---------------|----------------|-----|
| Type | MAC address 6 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomiface_vlan_domain

The name of the VLAN domain associated with the interface or interface port.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | # | Can be edited | Yes |

nomiface_vlan_number

The database identifier (ID) of the VLAN domain associated with the interface or interface port.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

nomiface_mac

The MAC address of the interface. If you specify it, this parameter overwrites **nomiface_port_mac**.

| | | | |
|---------------|-------------|----------------|-----|
| Type | MAC address | Maximum length | N/A |
| Default value | | Can be edited | Yes |

nomnetobj_id

The database identifier (ID) of the network object, a unique numeric key value automatically incremented when you add a network object. Use the ID to specify the network object of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomnetobj_name

The name of the network object. Use the name to specify the network object of your choice.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

nomfolder_path

The path to the Network Object Manager folder. Use the path to specify the folder of your choice.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

connected_port_id

The database identifier (ID) of the interface port you want to connect to the interface you are adding/editing.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

connected_port_name

The name of the interface port connected to the interface or interface port.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

connected_port_nomnetobj_id

The database identifier (ID) of the network object to which the interface port is connected.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

connected_port_nomfolder_path

The path to the folder of the network object to which the interface port is connected.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

connected_port_nomnetobj_name

The name of the network object to which the interface port is connected.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

nom_iface_list — List the interfaces

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

nomiface_id

The database identifier (ID) of the interface.

nomiface_fullname

The full name of the interface, its identifier, as follows: <nomiface_port_name> - <nomiface_vlan_number> - <nomiface_name> - <nomiface_hostaddr>. At the very least the identifier includes *Port* <name> - VLAN <ID>. If the interface is configured without a VLAN, the VLAN number is 0.

nomiface_name

The name of the interface.

nomport_id

The database identifier (ID) of the interface port.

nomiface_port_name

The name of the interface port.

nomiface_port_mac

The MAC address of the interface port, if no **nomiface_mac** is set.

nomiface_vlan_domain

The name of the VLAN domain associated with the interface.

nomiface_vlan_number

The database identifier (ID) of the VLAN domain associated with the interface.

nomiface_mac

The MAC address of the interface. This parameter overwrites **nomiface_port_mac**.

nomiface_main_mac

The MAC address of the interface, if it is set as Main interface. It matches **nomiface_port_mac** only if **nomiface_mac** is not specified.

nomiface_main

A way to determine if the interface is set as Main interface (1) or not (0).

nomiface_main_hostaddr

The human readable version of the Network Object Manager main interface IP address.

nomiface_ip_addr

The IPv4 address in hexadecimal format of the interface.

nomiface_hostaddr

The human readable version of the parameter **nomiface_ip_addr**.

nomiface_ip6_addr

The IPv6 address in hexadecimal format of the interface.

nomiface_hostaddr6

The human readable version of the parameter **nomiface_ip6_addr**.

nomnetobj_id

The database identifier (ID) of the network object the object belongs to.

nomnetobj_name

The name of the network object the object belongs to.

nomfolder_id

The database identifier (ID) of the Network Object Manager folder the object belongs to.

nomfolder_path

The path to the Network Object Manager folder the object belongs to.

nomfolder_name

The name of the Network Object Manager folder the object belongs to.

nomfolder_site_name

The name of the space associated with the Network Object Manager folder the object belongs to.

connected_port_id

The database identifier (ID) of the interface port connected to the interface.

connected_port_name

The name of the interface port connected to the interface.

connected_port_nomnetobj_id

The database identifier (ID) of the network object connected to the interface.

connected_port_nomnetobj_name

The name of the network object connected to the interface.

connected_port_nomfolder_path

The path to the folder of the network object connected to the interface.

ip_id

The database identifier (ID) of the IPv4 address.

ip_name

The name of the IPv4 address.

ip6_id

The database identifier (ID) of the IPv6 address.

ip6_name

The name of the IPv6 address.

ip46_name

The name of the matching IP address in the IPAM space associated with the folder the object belongs to.

nomnetobj_class_parameters

The class parameters applied to the network object the object belongs to and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&....

nomnetobj_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
nomnetobj_class_parameters: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....*

nomnetobj_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

nomfolder_class_parameters

The class parameters applied to the Network Object Manager folder the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

nomfolder_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **nomfolder_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

nom_iface_info — Display the properties of an interface

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

nomiface_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

nomiface_id

The database identifier (ID) of the interface, a unique numeric key value automatically incremented when you add an interface. Use the ID to specify the interface of your choice.

Output Parameters

nomiface_id

The database identifier (ID) of the interface.

nomiface_fullname

The full name of the interface, its identifier, as follows: <nomiface_port_name> - <nomiface_vlan_number> - <nomiface_name> - <nomiface_hostaddr>. At the very least the identifier includes Port <name> - VLAN <ID>. If the interface is configured without a VLAN, the VLAN number is 0.

nomiface_name

The name of the interface.

nomport_id

The database identifier (ID) of the interface port.

nomiface_port_name

The name of the interface port.

nomiface_port_mac

The MAC address of the interface port, if no **nomiface_mac** is set.

nomiface_vlan_domain

The name of the VLAN domain associated with the interface.

nomiface_vlan_number

The database identifier (ID) of the VLAN domain associated with the interface.

nomiface_mac

The MAC address of the interface. This parameter overwrites **nomiface_port_mac**.

nomiface_main_mac

The MAC address of the interface, if it is set as Main interface. It matches **nomiface_port_mac** only if **nomiface_mac** is not specified.

nomiface_main

A way to determine if the interface is set as Main interface (1) or not (0).

nomiface_main_hostaddr

The human readable version of the Network Object Manager main interface IP address.

nomiface_ip_addr

The IPv4 address in hexadecimal format of the interface.

nomiface_hostaddr

The human readable version of the parameter **nomiface_ip_addr**.

nomiface_ip6_addr

The IPv6 address in hexadecimal format of the interface.

nomiface_hostaddr6

The human readable version of the parameter **nomiface_ip6_addr**.

nomnetobj_id

The database identifier (ID) of the network object the object belongs to.

nomnetobj_name

The name of the network object the object belongs to.

nomfolder_id

The database identifier (ID) of the Network Object Manager folder the object belongs to.

nomfolder_path

The path to the Network Object Manager folder the object belongs to.

nomfolder_name

The name of the Network Object Manager folder the object belongs to.

nomfolder_site_name

The name of the space associated with the Network Object Manager folder the object belongs to.

connected_port_id

The database identifier (ID) of the interface port connected to the interface.

connected_port_name

The name of the interface port connected to the interface.

connected_port_nomnetobj_id

The database identifier (ID) of the network object connected to the interface.

connected_port_nomnetobj_name

The name of the network object connected to the interface.

connected_port_nomfolder_path

The path to the folder of the network object connected to the interface.

ip_id

The database identifier (ID) of the IPv4 address.

ip_name

The name of the IPv4 address.

ip6_id

The database identifier (ID) of the IPv6 address.

ip6_name

The name of the IPv6 address.

ip46_name

The name of the matching IP address in the IPAM space associated with the folder the object belongs to.

nomnetobj_class_parameters

The class parameters applied to the network object the object belongs to and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

nomnetobj_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
nomnetobj_class_parameters: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .*

nomnetobj_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:
<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

nomfolder_class_parameters

The class parameters applied to the Network Object Manager folder the object belongs to and their value: *<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .*

nomfolder_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
nomfolder_class_parameters: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .*

Name

nom_iface_count — Count the number of interfaces

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

nom_port_list — List the interface ports

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

nomport_id

The database identifier (ID) of the interface port.

nomport_name

The name of the interface port.

nomport_mac

The MAC address of the interface port. Upon addition, it was specified via **nomiface_port_mac**.

nomnetobj_id

The database identifier (ID) of the network object the object belongs to.

connected_port_id

The database identifier (ID) of the interface port connected to the interface port.

Name

nom_port_info — Display the properties of an interface port

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

nomport_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

nomport_id

The database identifier (ID) of the interface port, a unique numeric key value automatically incremented when you add an interface port. Use the ID to specify the interface port of your choice.

Output Parameters

nomport_id

The database identifier (ID) of the interface port.

nomport_name

The name of the interface port.

nomport_mac

The MAC address of the interface port. Upon addition, it was specified via **nom-interface_port_mac**.

nomnetobj_id

The database identifier (ID) of the network object the object belongs to.

connected_port_id

The database identifier (ID) of the interface port connected to the interface port.

Name

nom_port_count — Count the number of interface ports

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

nom_iface_delete — Delete an interface or an interface port

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

nomiface_id || (nomiface_port_name && (nomnetobj_id || (nomfolder_path && nomnetobj_name))

Input Parameters

nomiface_id

The database identifier (ID) of the interface or interface port, a unique numeric key value automatically incremented when you add an interface or interface port. Use the ID to specify the interface or interface port of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomiface_name

The name of the interface or interface port.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomiface_hostaddr

The IPv4 or IPv6 address of the interface or interface port.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomiface_port_name

The name of the interface port, as follows: *port_<number>*. Use the name to specify the interface or interface port of your choice.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

nomiface_vlan_domain

The name of the VLAN domain associated with the interface or interface port.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | # | Can be edited | Yes |

nomiface_vlan_number

The database identifier (ID) of the VLAN domain associated with the interface or interface port.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

nomnetobj_id

The database identifier (ID) of the network object, a unique numeric key value automatically incremented when you add a network object. Use the ID to specify the network object of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

nomnetobj_name

The name of the network object. Use the name to specify the network object of your choice.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

nomfolder_path

The path to the Network Object Manager folder. Use the path to specify the folder of your choice.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Part VI. Application Services

Table of Contents

| | |
|-------------------------------------|-----|
| 43. Application | 799 |
| app_application_add | 800 |
| app_application_list | 803 |
| app_application_info | 806 |
| app_application_count | 809 |
| app_application_groupby | 810 |
| app_application_groupby_count | 812 |
| app_application_delete | 814 |
| 44. Pool | 816 |
| app_pool_add | 817 |
| app_pool_list | 820 |
| app_pool_info | 823 |
| app_pool_count | 826 |
| app_pool_groupby | 827 |
| app_pool_groupby_count | 829 |
| app_pool_delete | 831 |
| 45. Node | 833 |
| app_node_add | 834 |
| app_node_list | 837 |
| app_node_info | 840 |
| app_node_count | 843 |
| app_node_groupby | 844 |
| app_node_groupby_count | 846 |
| app_node_delete | 848 |

Chapter 43. Application

Name

app_application_add — Add/Edit an application

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (name && fqdn)
- **Editing:** (appapplication_id || (name && fqdn))

Input Parameters

appapplication_id

The database identifier (ID) of the application, a unique numeric key value automatically incremented when you add an application. Use the ID to specify the application of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

name

The name of the application.

| Type | Regular expression: [-_\.\w-zA-Z0-9]+ | Maximum length | 128 |
|---------------|---------------------------------------|----------------|-----|
| Default value | N/A | Can be edited | No |

fqdn

The Fully Qualified Domain Name (FQDN) of the application.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | No |

gslbserver_list

The name of all the GSLB servers associated with the application. You can specify one or more names.

| Type | List of strings separated by ; | Maximum length | N/A |
|---------------|--------------------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

appapplication_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

appapplication_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

appapplication_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... . If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
|------|--------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

app_application_list — List the applications

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

tree_level

The database level of the server associated with the application. *0* indicates the server is managed on its own, *1* indicates it is managed by a smart architecture.

tree_path

The database path toward the server associated with the application, as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

dns_type

The type of DNS server associated with the application. It is only returned for deployed applications.

appapplication_id

The database identifier (ID) of the application.

appapplication_gslbserver_name

The name of the GSLB server associated with the application. It is only returned for deployed applications.

appapplication_name

The name of the application.

appapplication_fqdn

The FQDN of the application.

appapplication_gslbserver_id

The database identifier (ID) of the GSLB server associated with the application. It is only returned for deployed applications.

parent_application_id

The database identifier (ID) of the application. It is only returned for deployed applications.

parent_application_name

The name of the application. It is only returned for deployed applications.

delayed_time

The delay of creation/deletion status. *1* indicates that the object is not created/deleted yet.

appapplication_gslbserver_status

The status of the GSLB server associated with the application, either OK (*1*), GSLB Stopped (*2*), GSLB Invalid Credentials (*4*) or GSLB Timeout (*5*). It is only returned for deployed applications.

appapplication_total_nodes

The number of nodes of the application.

appapplication_inactive_nodes

The number of nodes of the application that are *Inactive* (down).

appapplication_gslbserver_list

The name of all the GSLB servers associated with the application. It lists the name of each server separated by a comma.

appapplication_class_name

The name of the class applied to the object, it can be preceded by the class directory.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 43.1. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

appapplication_class_parameters

The class parameters applied to the application and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

appapplication_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **appapplication_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

appapplication_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

Name

app_application_info — Display the properties of an application

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

appapplication_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

appapplication_id

The database identifier (ID) of the application, a unique numeric key value automatically incremented when you add an application. Use the ID to specify the application of your choice.

Output Parameters

tree_level

The database level of the server associated with the application. 0 indicates the server is managed on its own, 1 indicates it is managed by a smart architecture.

tree_path

The database path toward the server associated with the application, as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

dns_type

The type of DNS server associated with the application. It is only returned for deployed applications.

appapplication_id

The database identifier (ID) of the application.

appapplication_gslbserver_name

The name of the GSLB server associated with the application. It is only returned for deployed applications.

appapplication_name

The name of the application.

appapplication_fqdn

The FQDN of the application.

appapplication_gslbserver_id

The database identifier (ID) of the GSLB server associated with the application. It is only returned for deployed applications.

parent_application_id

The database identifier (ID) of the application. It is only returned for deployed applications.

parent_application_name

The name of the application. It is only returned for deployed applications.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

appapplication_gslbserver_status

The status of the GSLB server associated with the application, either OK (1), GSLB Stopped (2), GSLB Invalid Credentials (4) or GSLB Timeout (5). It is only returned for deployed applications.

appapplication_total_nodes

The number of nodes of the application.

appapplication_inactive_nodes

The number of nodes of the application that are *Inactive* (down).

appapplication_gslbserver_list

The name of all the GSLB servers associated with the application. It lists the name of each server separated by a comma.

appapplication_class_name

The name of the class applied to the object, it can be preceded by the class directory.

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 43.2. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

appapplication_class_parameters

The class parameters applied to the application and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

appapplication_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **appapplication_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

appapplication_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

app_application_count — Count the number of applications

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

app_application_groupby — Group applications by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

app_application_groupby_count — Count the number of applications grouped by parameter(s)

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

app_application_delete — Delete an application

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(appapplication_id || (name && fqdn))

Input Parameters**appapplication_id**

The database identifier (ID) of the application, a unique numeric key value automatically incremented when you add an application. Use the ID to specify the application of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

name

The name of the application.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

fqdn

The Fully Qualified Domain Name (FQDN) of the application.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

gslbserver_id

The database identifier (ID) of the GSLB server associated with the application, a unique numeric key value automatically incremented when you add the server. Use it to identify the GSLB server of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

gslbserver_name

The name of the GSLB server associated with the application.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 44. Pool

Name

app_pool_add — Add/Edit a pool

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (name && type && lb_mode && (appapplication_id || (appapplication_name && appapplication_fqdn)))
- **Editing:** (apppool_id || (name && (appapplication_id || (appapplication_name && appapplication_fqdn))))

Input Parameters

apppool_id

The database identifier (ID) of the pool, a unique numeric key value automatically incremented when you add a pool. Use the ID to specify the pool of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

appapplication_id

The database identifier (ID) of the application the object belongs to.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

appapplication_name

The name of the application the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

appapplication_fqdn

The Fully Qualified Domain Name (FQDN) of the application the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

appapplication_gslbserver_id

The database identifier (ID) of the GSLB server associated with the application, a unique numeric key value automatically incremented when you add the server. Use it to identify the GSLB server of your choice.

Pool

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

name

The name of the pool.

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Regular expression: [-_\.\,a-zA-Z0-9]+ | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

type

The type of the pool.

| | | | |
|----------------------|---------------------------|-----------------------|-----|
| Type | Fixed value: ipv4 ipv6 | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

lb_mode

The load-balancing mode of the pool.

| | | | |
|----------------------|---|-----------------------|-----|
| Type | Fixed value: weighted round-robin latency | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

best_active_nodes

The maximum number of active nodes with the lowest latency that must answer the queries made to the application FQDN. You only need to set it if you set the **lb_mode** to *latency*.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

affinity_state

The session affinity activation status.

| | | | |
|----------------------|-----------------|-----------------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

affinity_session_time

The session duration, in seconds. You only need to set it if the parameter **affinity_state** is set to 1.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

app_pool_list — List the pools

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

tree_level

The database level of the server associated with the application the object belongs to. 0 indicates the server is managed on its own, 1 indicates it is managed by a smart architecture.

tree_path

The database path toward the server associated with the application the object belongs to, as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

dns_type

The type of DNS server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_gslbserver_name

The name of the GSLB server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_id

The database identifier (ID) of the application the object belongs to.

appapplication_gslbserver_id

The database identifier (ID) of the GSLB server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_name

The name of the application the object belongs to.

appapplication_fqdn

The FQDN of the application the object belongs to.

parent_application_id

The database identifier (ID) of the application the object belongs to. It is only returned for deployed applications.

apppool_id

The database identifier (ID) of the pool.

apppool_name

The name of the pool.

apppool_type

The type of the pool, *ipv4* or *ipv6*.

apppool_weight

Not documented. Internal use.

apppool_lb_mode

The load-balancing mode of the pool, either *weighted*, *round-robin* or *latency*.

translated_apppool_lb_mode

The load-balancing mode of the pool, as displayed in the GUI.

apppool_best_active_nodes

The maximum number of active nodes with the lowest latency that must answer the queries made to the application FQDN. It is only returned if the parameter **apppool_lb_mode** is set to *latency*.

apppool_affinity_state

The session affinity activation status.

apppool_affinity_session_time

The session duration, in seconds. It is only returned if the parameter **affinity_state** is set to 1.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

apppool_total_nodes

The number of nodes of the pool.

apppool_inactive_nodes

The number nodes of the pool that are *Inactive* (down).

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 44.1. Multi-status severity levels

| Message number | Severity | Description |
|----------------|---------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

Name

app_pool_info — Display the properties of a pool

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

apppool_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

apppool_id

The database identifier (ID) of the pool, a unique numeric key value automatically incremented when you add a pool. Use the ID to specify the pool of your choice.

Output Parameters

tree_level

The database level of the server associated with the application the object belongs to. 0 indicates the server is managed on its own, 1 indicates it is managed by a smart architecture.

tree_path

The database path toward the server associated with the application the object belongs to, as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

dns_type

The type of DNS server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_gslbserver_name

The name of the GSLB server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_id

The database identifier (ID) of the application the object belongs to.

appapplication_gslbserver_id

The database identifier (ID) of the GSLB server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_name

The name of the application the object belongs to.

appapplication_fqdn

The FQDN of the application the object belongs to.

parent_application_id

The database identifier (ID) of the application the object belongs to. It is only returned for deployed applications.

apppool_id

The database identifier (ID) of the pool.

apppool_name

The name of the pool.

apppool_type

The type of the pool, *ipv4* or *ipv6*.

apppool_weight

Not documented. Internal use.

apppool_lb_mode

The load-balancing mode of the pool, either *weighted*, *round-robin* or *latency*.

translated_apppool_lb_mode

The load-balancing mode of the pool, as displayed in the GUI.

apppool_best_active_nodes

The maximum number of active nodes with the lowest latency that must answer the queries made to the application FQDN. It is only returned if the parameter **apppool_lb_mode** is set to *latency*.

apppool_affinity_state

The session affinity activation status.

apppool_affinity_session_time

The session duration, in seconds. It is only returned if the parameter **affinity_state** is set to 1.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

apppool_total_nodes

The number of nodes of the pool.

apppool_inactive_nodes

The number nodes of the pool that are *Inactive* (down).

multistatus

The Multi-status information is displayed as follows: <number-of-instances>@<message-number>@<multi-status-severity>@<module>. The different severity levels are:

Table 44.2. Multi-status severity levels

| Message number | Severity | Description |
|-----------------------|-----------------|--|
| 0 - 16 | Emergency | The object configuration prevents the system from running properly. Action is required. |
| 17 - 33 | Critical | The object configuration is in critical conditions. Immediate action is recommended. |
| 34 - 50 | Error | The object configuration failed at some level. Action is recommended. |
| 51 - 66 | Warning | The object configuration triggers error messages if no action is taken. Action to be taken at your discretion. |
| 67 - 83 | Notice | The object configuration is normal but undergoing events that might trigger errors. No immediate action required. |
| 84 - 100 | Informational | The object configuration is normal, operational messages (might inform you about potential incompatibilities with other modules, etc). No action required. |

Name

app_pool_count — Count the number of pools

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

app_pool_groupby — Group pools by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

app_pool_groupby_count — Count the number of pools grouped by parameter(s)

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>'* or *<parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

app_pool_delete — Delete a pool

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(apppool_id || (name && (appapplication_id || (appapplication_name && appapplication_fqdn))))

Input Parameters

apppool_id

The database identifier (ID) of the pool, a unique numeric key value automatically incremented when you add a pool. Use the ID to specify the pool of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

name

The name of the pool.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

appapplication_id

The database identifier (ID) of the application the object belongs to.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

appapplication_name

The name of the application the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

appapplication_fqdn

The Fully Qualified Domain Name (FQDN) of the application the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

appapplication_gslbserver_id

The database identifier (ID) of the GSLB server associated with the application, a unique numeric key value automatically incremented when you add the server. Use it to identify the GSLB server of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 45. Node

Name

app_node_add — Add/Edit a node

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (name && hostaddr && (apppool_id || (apppool_name && (appapplication_id || (appapplication_name && appapplication_fqdn)))))
- **Editing:** (appnode_id || (name && (apppool_id || (apppool_name && (appapplication_id || (appapplication_name && appapplication_fqdn))))))

Input Parameters

appnode_id

The database identifier (ID) of the node, a unique numeric key value automatically incremented when you add a node. Use the ID to specify the node of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

appapplication_id

The database identifier (ID) of the application the object belongs to.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

appapplication_name

The name of the application the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

appapplication_fqdn

The Fully Qualified Domain Name (FQDN) of the application the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

appapplication_gslbserver_id

The database identifier (ID) of the GSLB server associated with the application, a unique numeric key value automatically incremented when you add the server. Use it to identify the GSLB server of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

apppool_id

The database identifier (ID) of the pool, a unique numeric key value automatically incremented when you add a pool. Use the ID to specify the pool of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

apppool_name

The name of the pool.

| | | | |
|---------------|--|----------------|-----|
| Type | Regular expression: [-_\.\,a-zA-Z0-9]+ | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

name

The name of the node.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

hostaddr

The IPv4 or IPv6 address of the node.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

weight

The weight of the node, it sets which node is used first within the pool. It must be an integer between 0 and 255, where 0 sets a node as backup.

| | | | |
|---------------|---|----------------|-----|
| Type | Regular expression: ^([0-9][[1-8][0-9] 9[0-9]] 1[0-9]{2} 2[0-4][0-9] 25[0-5])\$ | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

admin_status

The administrative status of the node, managed (1) or unmanaged (0).

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

status

Internal use. Not documented.

| | | | |
|---------------|--------------------------|----------------|-----|
| Type | Fixed value: 0 1 2 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

apphealthcheck_name

The type of health check of the node.

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: ok ping tcp http custom | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

app_node_list — List the nodes

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

tree_level

The database level of the server associated with the application the object belongs to. 0 indicates the server is managed on its own, 1 indicates it is managed by a smart architecture.

tree_path

The database path toward the server associated with the application the object belongs to, as follows: <server-name>#. If the physical server is managed through a smart architecture, the path looks as follows: <smart-architecture-name>##<server-name>.

dns_type

The type of DNS server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_gslbserver_name

The name of the GSLB server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_gslbserver_id

The database identifier (ID) of the GSLB server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_id

The database identifier (ID) of the application the object belongs to.

appapplication_name

The name of the application the object belongs to.

appapplication_fqdn

The FQDN of the application the object belongs to.

parent_application_id

The database identifier (ID) of the application the object belongs to. It is only returned for deployed applications.

apppool_id

The database identifier (ID) of the pool.

apppool_name

The name of the pool.

apppool_lb_mode

The load-balancing mode of the pool the object belongs to, either *weighted*, *round-robin* or *latency*.

translated_apppool_lb_mode

The load-balancing mode of the pool the object belongs to, as displayed in the GUI.

appnode_id

The database identifier (ID) of the node.

appnode_name

The name of the node.

appnode_ip_addr

The IPv4 address of the node.

appnode_ip6_addr

The IPv6 address of the node.

appnode_weight

The weight of the node, an integer between 0 and 255. 0 indicates a backup node.

admin_status

The administrative status of the node. 1 indicates the node is managed. 0 indicates that is unmanaged and ignored.

appnode_status

The operational status of the node:

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

apphealthcheck_name

The type of health check configured for the node.

translated_apphealthcheck_name

The health check type, as displayed in the GUI.

apphealthcheck_id

The database identifier (ID) of the health check configured for the node.

apphealthcheck_freq

The frequency to which the health check configured for the node is performed, in seconds.

apphealthcheck_failover

The number of times, between 1 and 10, before the parameter **appnode_status** is set to 0 (inactive) due to the health check result. By default, it is set to 3.

apphealthcheck_failback

The number of times, between 1 and 10, before the parameter **appnode_status** is set to 1 (active) due to the health check result. By default, it is set to 3.

apphealthcheck_timeout

The number of seconds, between 1 and 10, before the health check times out if the node is not responding.

apphealthcheck_params

The rest of the health check parameters configured, when relevant.

Name

app_node_info — Display the properties of a node

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

appnode_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

appnode_id

The database identifier (ID) of the node, a unique numeric key value automatically incremented when you add a node. Use the ID to specify the node of your choice.

Output Parameters

tree_level

The database level of the server associated with the application the object belongs to. `0` indicates the server is managed on its own, `1` indicates it is managed by a smart architecture.

tree_path

The database path toward the server associated with the application the object belongs to, as follows: `<server-name>#`. If the physical server is managed through a smart architecture, the path looks as follows: `<smart-architecture-name>##<server-name>`.

dns_type

The type of DNS server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_gslbserver_name

The name of the GSLB server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_gslbserver_id

The database identifier (ID) of the GSLB server associated with the application the object belongs to. It is only returned for deployed applications.

appapplication_id

The database identifier (ID) of the application the object belongs to.

appapplication_name

The name of the application the object belongs to.

appapplication_fqdn

The FQDN of the application the object belongs to.

parent_application_id

The database identifier (ID) of the application the object belongs to. It is only returned for deployed applications.

apppool_id

The database identifier (ID) of the pool.

apppool_name

The name of the pool.

apppool_lb_mode

The load-balancing mode of the pool the object belongs to, either *weighted*, *round-robin* or *latency*.

translated_apppool_lb_mode

The load-balancing mode of the pool the object belongs to, as displayed in the GUI.

appnode_id

The database identifier (ID) of the node.

appnode_name

The name of the node.

appnode_ip_addr

The IPv4 address of the node.

appnode_ip6_addr

The IPv6 address of the node.

appnode_weight

The weight of the node, an integer between 0 and 255. 0 indicates a backup node.

admin_status

The administrative status of the node. 1 indicates the node is managed. 0 indicates that is unmanaged and ignored.

appnode_status

The operational status of the node:

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

apphealthcheck_name

The type of health check configured for the node.

translated_apphealthcheck_name

The health check type, as displayed in the GUI.

apphealthcheck_id

The database identifier (ID) of the health check configured for the node.

apphealthcheck_freq

The frequency to which the health check configured for the node is performed, in seconds.

apphealthcheck_failover

The number of times, between 1 and 10, before the parameter **appnode_status** is set to 0 (inactive) due to the health check result. By default, it is set to 3.

apphealthcheck_failback

The number of times, between 1 and 10, before the parameter **appnode_status** is set to 1 (active) due to the health check result. By default, it is set to 3.

apphealthcheck_timeout

The number of seconds, between 1 and 10, before the health check times out if the node is not responding.

apphealthcheck_params

The rest of the health check parameters configured, when relevant.

Name

app_node_count — Count the number of nodes

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

app_node_groupby — Group nodes by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: `GROUPBY=<param1>,<param2>,...`. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement **SELECT** is returned.

Name

app_node_groupby_count — Count the number of node grouped by parameter(s)

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>'* or *<parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

app_node_delete — Delete a node

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(appnode_id || (name && (apppool_id || (apppool_name && (appapplication_id || (appapplication_name && appapplication_fqdn))))))

Input Parameters

appnode_id

The database identifier (ID) of the node, a unique numeric key value automatically incremented when you add a node. Use the ID to specify the node of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

name

The name of the node.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

apppool_id

The database identifier (ID) of the pool, a unique numeric key value automatically incremented when you add a pool. Use the ID to specify the pool of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

apppool_name

The name of the pool.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

appapplication_id

The database identifier (ID) of the application the object belongs to.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

appapplication_name

The name of the application the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

appapplication_fqdn

The Fully Qualified Domain Name (FQDN) of the application the object belongs to.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

appapplication_gslbserver_id

The database identifier (ID) of the GSLB server associated with the application, a unique numeric key value automatically incremented when you add the server. Use it to identify the GSLB server of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Part VII. Guardian Services

Table of Contents

| | |
|-------------------------------------|-----|
| 46. Policy | 852 |
| guardian_policy_add | 853 |
| guardian_policy_list | 855 |
| guardian_policy_info | 857 |
| guardian_policy_count | 859 |
| guardian_policy_groupby | 860 |
| guardian_policy_groupby_count | 862 |
| guardian_policy_delete | 864 |

Chapter 46. Policy

Name

guardian_policy_add — Add/Edit a policy

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (name)
- **Editing:** (guardianpolicy_id || (name))

Input Parameters

guardianpolicy_id

The database identifier (ID) of the policy, a unique numeric key value automatically incremented when you add the policy. Use it to identify the policy of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

name

The name of the policy.

| Type | Regular expression: [-\.\w-zA-Z0-9]+ | Maximum length | 40 |
|---------------|--------------------------------------|----------------|----|
| Default value | N/A | Can be edited | No |

description

The description of the policy.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dns_list

The name of all the Guardian servers associated with the policy. You can specify one or more names.

| Type | List of strings separated by ; | Maximum length | N/A |
|---------------|--------------------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

guardian_policy_list — List the policies

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

guardianpolicy_id

The database identifier (ID) of the policy.

guardianpolicy_name

The name of the policy.

guardianpolicy_description

The description of the policy.

guardianpolicy_READONLY

The read only status of the policy. If set to 1, the policy cannot be edited.

dns_id

The database identifier (ID) of the Guardian server associated with the policy. It is only returned for deployed policies.

dns_name

The name of the Guardian server associated with the policy. It is only returned for deployed policies.

parent_guardianpolicy_id

The database identifier (ID) of the policy. It is only returned for deployed policies.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

Name

guardian_policy_info — Display the properties of a policy

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

guardianpolicy_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

guardianpolicy_id

The database identifier (ID) of the policy, a unique numeric key value automatically incremented when you add the policy. Use it to identify the policy of your choice.

Output Parameters

guardianpolicy_id

The database identifier (ID) of the policy.

guardianpolicy_name

The name of the policy.

guardianpolicy_description

The description of the policy.

guardianpolicy_READONLY

The read only status of the policy. If set to 1, the policy cannot be edited.

dns_id

The database identifier (ID) of the Guardian server associated with the policy. It is only returned for deployed policies.

dns_name

The name of the Guardian server associated with the policy. It is only returned for deployed policies.

parent_guardianpolicy_id

The database identifier (ID) of the policy. It is only returned for deployed policies.

delayed_time

The delay of creation/deletion status. 1 indicates that the object is not created/deleted yet.

Name

guardian_policy_count — Count the number of policies

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

guardian_policy_groupby — Group nodes by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

guardian_policy_groupby_count — Count the number of policies grouped by parameter(s)

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

guardian_policy_delete — Delete a policy

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(guardianpolicy_id || (name && (dns_name || dns_id)))

Input Parameters

guardianpolicy_id

The database identifier (ID) of the policy, a unique numeric key value automatically incremented when you add the policy. Use it to identify the policy of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

name

The name of the policy.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 40 |
| Default value | N/A | Can be edited | Yes |

dns_id

The database identifier (ID) of the Guardian server associated with the policy, a unique numeric key value automatically incremented when you add the server. Use it to identify the Guardian server of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

dns_name

The name of the Guardian server associated with the policy.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Part VIII. Cloud Observer Services

Table of Contents

| | |
|-------------------------------------|-----|
| 47. Plugin | 868 |
| co_plugin_list | 869 |
| 48. Worker | 871 |
| co_worker_add | 872 |
| co_worker_list | 876 |
| co_worker_info | 880 |
| co_worker_count | 883 |
| co_worker_delete | 884 |
| 49. Cloud Observer Folder | 886 |
| co_folder_list | 887 |
| co_folder_info | 890 |
| co_folder_count | 893 |
| 50. Cloud Observer Instance | 894 |
| co_instance_list | 895 |
| co_instance_info | 898 |
| co_instance_count | 901 |
| co_instance_network_list | 902 |
| co_instance_network_info | 906 |
| co_instance_network_count | 909 |
| 51. Cloud Observer Network | 910 |
| co_network_list | 911 |
| co_network_info | 914 |
| co_network_count | 917 |
| co_network_instance_list | 918 |
| co_network_instance_info | 922 |
| co_network_instance_count | 925 |
| 52. Cloud Observer IP Address | 926 |
| co_ip_list | 927 |
| co_ip_info | 931 |
| co_ip_count | 934 |
| co_ip_log_list | 935 |
| co_ip_log_info | 938 |
| co_ip_log_count | 940 |

Chapter 47. Plugin

Name

co_plugin_list — List the plugins

Description

This service allows you to list the Cloud Observer plugins and display what information is required or not to add a worker via the service *co_worker_add*.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...* .

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

coplugin_id

The database identifier (ID) of the plugin.

coplugin_name

The shortname of the cloud provider of the plugin, either *Azure* for Microsoft Azure data, *Ec2* for Amazon AWS data, *GCP* for Google Cloud Platform data, *vCenter* for VMware vCenter data or *Meraki* for Cisco Meraki.

coplugin_label

The name of the cloud provider of the plugin, either *Microsoft Azure*, *Amazon EC2*, *Google Cloud Platform*, *VMWare vCenter* or *Cisco Meraki*.

coplugin_version

The version number of the plugin.

coplugin_params

The expected parameters of the plugin used to add a worker. The list of parameters is unique to each **coplugin_name**, it is returned in JSON format and indicates if each parameter is required or not.

coplugin_demo

Internal use. Not documented.

Chapter 48. Worker

Name

co_worker_add — Add/Edit a worker

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (coworker_name && (coplugin_id || coplugin_name))
- **Editing:** (coworker_id || coworker_uuid || coworker_name)

Input Parameters

coworker_id

The database identifier (ID) of the worker, a unique numeric key value automatically incremented when you add a worker. Use the ID to specify the worker of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

coworker_uuid

The universally unique identifier (UUID) of the worker, a unique numeric key value automatically generated when you add a worker. Use the UUID to specify the worker of your choice.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

coworker_name

The name of the worker, each worker must have a unique name. Use the name to specify which worker to edit.

| Type | String | Maximum length | 64 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

coworker_descr

The description of the worker.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

coworker_version

The version number of the plugin used by the worker, as follows: x.x.x .

| Type | String | Maximum length | 32 |
|------|--------|----------------|----|
|------|--------|----------------|----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

coplugin_id

The database identifier (ID) of the cloud provider of the plugin, a unique numeric key value automatically incremented when you add a plugin. Use the ID to specify the plugin of your choice.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

coplugin_name

The shortname of the cloud provider of the plugin. Use the shortname to specify the cloud provider of your choice, either *Azure* for Microsoft Azure data, *Ec2* for Amazon AWS data, *GCP* for Google Cloud Platform data, *vCenter* for VMware vCenter data or *Meraki* for Cisco Meraki data.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

member_id

The database identifier (ID) of the remote SOLIDserver appliance on which you want to run the worker.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of the space associated with the worker, a numeric value automatically incremented when you add a space. Use the ID to specify the worker of your choice.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

coworker_refresh_parameters

A way to configure a scheduled refresh of the database of the worker you are adding/editing. The refresh frequency must be specified following the format *<minute>:<hour>:<date-of-the-month>:<month>:<day-of-the-week>* where:

- *<minute>* accepts values between 00 and 59.
- *<hour>* accepts values between 00 and 12.
- *<date-of-the-month>* accepts values between 1 and 31.
- *<month>* accepts values between 1 and 12, where 1 is January, etc.
- *<day-of-the-week>* accepts values between 1 and 7, where 1 is Monday, etc.

All periods must be specified and separated by : . Each period also accepts a valid regex pattern for datetime. Note that . * indicates any value of a period and allows, for instance, to set a refresh once a year on December 21st at 0h00 with 00:00:21:12:. * .

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

coworker_config

The configuration parameters of the worker you are adding/editing, in JSON format. The list of expected parameters varies for each plugin, you can obtain it thanks to the service `co_plugin_list`.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

coworker_enabled

The activation status of the worker. If set to `1` it is enabled, if set to `0` it is disabled. A disabled worker stops retrieving or updating new data, it cannot be refreshed.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

connectionprofile_id

The database identifier (ID) of the connection profile used as connection method, a numeric value automatically incremented when you add a connection profile. Use the ID to specify the connection profile of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

proxyprofile_id

The database identifier (ID) of the connection profile used as proxy connection method, a numeric value automatically incremented when you add a proxy connection profile. Use the ID to specify the proxy connection profile of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

connectionprofile_name

The name of the connection profile used as connection method.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

proxyprofile_name

The name of the connection profile used as proxy connection method.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (`new_only`) or if you want to edit an existing object but not create a new one (`edit_only`).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

coworker_uuid

The universally unique identifier (UUID) of the worker.

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

co_worker_list — List the workers

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

coplugin_id

The database identifier (ID) of the plugin used by the worker.

coplugin_name

The shortname of the cloud provider of the plugin used by the worker, either *Azure* for Microsoft Azure data, *Ec2* for Amazon AWS data, *GCP* for Google Cloud Platform data, *vCenter* for VMWare vCenter data or *Meraki* for Cisco Meraki data..

coplugin_label

The name of the cloud provider of the plugin used by the worker, either *Microsoft Azure*, *Amazon EC2*, *Google Cloud Platform* or *VMWare vCenter* or *Cisco Meraki*.

coplugin_version

The version number of the plugin used by the worker.

coplugin_params

The parameters of the plugin used by the worker.

coworker_id

The database identifier (ID) of the worker.

coworker_uuid

The universally unique identifier (UUID) of the worker.

coworker_name

The name of the worker.

coworker_descr

The description of the worker.

coworker_version

The version number of the plugin used by the worker.

coworker_instances

The number of instances retrieved by the worker.

coworker_ips

The number of IP addresses retrieved by the worker.

coworker_lastruntime

The last time the worker has been executed, in decimal UNIX date format.

coworker_updatetime

The last time any data regarding the worker was received, in decimal UNIX date format.

coworker_duration

The duration of the last worker refresh, in seconds.

coworker_size

The size of the file retrieved during the last worker refresh, in bytes.

coworker_size_kb

The size of the file retrieved during the last worker refresh, in kilobytes.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker.

coworker_config

The configuration parameters of the worker, in JSON format.

coworker_enabled

The activation status of the worker, either enabled (1) or disabled (0).

coworker_status

The operational status of the worker, either Unknown (0), OK (1), Plugin error (5), Plugin Missing (6), Provider error (E), Timeout (7), Insufficient privileges (P) or Invalid credentials (I).

member_id

The database identifier (ID) of the remote SOLIDserver appliance the worker is running on, 0 indicates the Local SOLIDserver.

member_name

The name of the remote SOLIDserver appliance the worker is running on.

site_id

The database identifier (ID) of the space associated with the worker.

site_name

The name of the space associated with the worker.

connectionprofile_id

The database identifier (ID) of the connection profile used as connection method for the worker, a numeric value automatically incremented when you add a connection profile.

proxyprofile_id

The database identifier (ID) of the connection profile used as proxy connection method for the worker, a numeric value automatically incremented when you add a proxy connection profile.

connectionprofile_name

The name of the connection profile used as connection method for the worker.

proxyprofile_name

The name of the connection profile used as proxy connection method for the worker.

connectionprofile_params

The connection parameters of the connection profile used as connection method for the worker.

connectionprofile_identifier

The third-party identifier of the connection profile used as connection method for the worker, either *Azure application ID* for Microsoft Azure, *Service account email* for Google Cloud Platform, *API key* for Cisco Meraki, *Login* for VMware vCenter or *Access Key ID* for Amazon Ec2.

connectionprofile_secret

The third-party secret of the connection profile used as connection method for the worker, either *Azure secret* for Microsoft Azure, *Service account private key* for Google Cloud Platform, *Password* for VMware vCenter or *Access secret key* for Amazon Ec2. Cisco Meraki does not have a secret.

coworker_class_parameters

The class parameters applied to the worker and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

coworker_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **coworker_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

coworker_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_worker_info — Display the properties of a worker

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

coworker_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

coworker_id

The database identifier (ID) of the worker, a unique numeric key value automatically incremented when you add a worker. Use the ID to specify the worker of your choice.

Output Parameters

coplugin_id

The database identifier (ID) of the plugin used by the worker.

coplugin_name

The shortname of the cloud provider of the plugin used by the worker, either *Azure* for Microsoft Azure data, *Ec2* for Amazon AWS data, *GCP* for Google Cloud Platform data, *vCenter* for VMWare vCenter data or *Meraki* for Cisco Meraki data..

coplugin_label

The name of the cloud provider of the plugin used by the worker, either *Microsoft Azure*, *Amazon EC2*, *Google Cloud Platform* or *VMWare vCenter* or *Cisco Meraki*.

coplugin_version

The version number of the plugin used by the worker.

coplugin_params

The parameters of the plugin used by the worker.

coworker_id

The database identifier (ID) of the worker.

coworker_uuid

The universally unique identifier (UUID) of the worker.

coworker_name

The name of the worker.

coworker_descr

The description of the worker.

coworker_version

The version number of the plugin used by the worker.

coworker_instances

The number of instances retrieved by the worker.

coworker_ips

The number of IP addresses retrieved by the worker.

coworker_lastruntime

The last time the worker has been executed, in decimal UNIX date format.

coworker_updatetime

The last time any data regarding the worker was received, in decimal UNIX date format.

coworker_duration

The duration of the last worker refresh, in seconds.

coworker_size

The size of the file retrieved during the last worker refresh, in bytes.

coworker_size_kb

The size of the file retrieved during the last worker refresh, in kilobytes.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker.

coworker_config

The configuration parameters of the worker, in JSON format.

coworker_enabled

The activation status of the worker, either enabled (1) or disabled (0).

coworker_status

The operational status of the worker, either Unknown (0), OK (1), Plugin error (5), Plugin Missing (6), Provider error (E), Timeout (T), Insufficient privileges (P) or Invalid credentials (I).

member_id

The database identifier (ID) of the remote SOLIDserver appliance the worker is running on, 0 indicates the Local SOLIDserver.

member_name

The name of the remote SOLIDserver appliance the worker is running on.

site_id

The database identifier (ID) of the space associated with the worker.

site_name

The name of the space associated with the worker.

connectionprofile_id

The database identifier (ID) of the connection profile used as connection method for the worker, a numeric value automatically incremented when you add a connection profile.

proxyprofile_id

The database identifier (ID) of the connection profile used as proxy connection method for the worker, a numeric value automatically incremented when you add a proxy connection profile.

connectionprofile_name

The name of the connection profile used as connection method for the worker.

proxyprofile_name

The name of the connection profile used as proxy connection method for the worker.

connectionprofile_params

The connection parameters of the connection profile used as connection method for the worker.

connectionprofile_identifier

The third-party identifier of the connection profile used as connection method for the worker, either *Azure application ID* for Microsoft Azure, *Service account email* for Google Cloud Platform, *API key* for Cisco Meraki, *Login* for VMware vCenter or *Access Key ID* for Amazon Ec2.

connectionprofile_secret

The third-party secret of the connection profile used as connection method for the worker, either *Azure secret* for Microsoft Azure, *Service account private key* for Google Cloud Platform, *Password* for VMware vCenter or *Access secret key* for Amazon Ec2. Cisco Meraki does not have a secret.

coworker_class_parameters

The class parameters applied to the worker and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

coworker_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **coworker_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

coworker_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_worker_count — Count the number of workers

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

co_worker_delete — Delete a worker

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(coworker_id || coworker_uuid)

Input Parameters

coworker_id

The database identifier (ID) of the worker, a unique numeric key value automatically incremented when you add a worker. Use the ID to specify the worker of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

coworker_uuid

The universally unique identifier (UUID) of the worker, a unique numeric key value automatically generated when you add a worker. Use the UUID to specify the worker of your choice.

| Type | String | Maximum length | 64 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

connectionprofile_id

The database identifier (ID) of the connection profile used as connection method, a numeric value automatically incremented when you add a connection profile. Use the ID to specify the connection profile of your choice.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

proxyprofile_id

The database identifier (ID) of the connection profile used as proxy connection method, a numeric value automatically incremented when you add a proxy connection profile. Use the ID to specify the proxy connection profile of your choice.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|------|---------------------|----------------|-----|
|------|---------------------|----------------|-----|

| | | | |
|---------------|-----|---------------|-----|
| Default value | N/A | Can be edited | Yes |
|---------------|-----|---------------|-----|

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 49. Cloud Observer Folder

Name

co_folder_list — List the Cloud Observer folders

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

cofolder_id

The database identifier (ID) of the folder.

cofolder_name

The name of the folder.

coworker_id

The database identifier (ID) of the worker that retrieved the folder.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder.

cofolder_instances

The number of instances of the folder.

cofolder_ips

The number of IP addresses of the folder.

cofolder_firstseen

The first time the folder was seen, in decimal UNIX date format.

cofolder_updatetime

The last time any data regarding the folder was received, in decimal UNIX date format.

cofolder_laststatustime

The last time the status of the folder has been refreshed, in decimal UNIX date format.

cofolder_statusperiod

The folder status duration, in seconds.

cofolder_status

The status of the folder.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder.

coworker_name

The name of the worker that retrieved the folder.

coworker_version

The version number of the plugin used by the worker that retrieved the folder.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder.

coworker_config

The configuration parameters of the worker that retrieved the folder, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder.

copugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder.

copugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder.

copugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder.

copugin_version

The version number of the plugin used by the worker that retrieved the folder.

copugin_params

The parameters of the plugin used by the worker that retrieved the folder.

cofolder_class_parameters

The class parameters applied to the Cloud Observer folder and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

cofolder_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **cofolder_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

cofolder_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:

<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_folder_info — Display the properties of a Cloud Observer folder

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

cofolder_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

cofolder_id

The database identifier (ID) of the folder, a unique numeric key value automatically incremented when a folder is retrieved. Use the ID to specify the folder of your choice.

Output Parameters

cofolder_id

The database identifier (ID) of the folder.

cofolder_name

The name of the folder.

coworker_id

The database identifier (ID) of the worker that retrieved the folder.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder.

cofolder_instances

The number of instances of the folder.

cofolder_ips

The number of IP addresses of the folder.

cofolder_firstseen

The first time the folder was seen, in decimal UNIX date format.

cofolder_updatetime

The last time any data regarding the folder was received, in decimal UNIX date format.

cofolder_laststatustime

The last time the status of the folder has been refreshed, in decimal UNIX date format.

cofolder_statusperiod

The folder status duration, in seconds.

cofolder_status

The status of the folder.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder.

coworker_name

The name of the worker that retrieved the folder.

coworker_version

The version number of the plugin used by the worker that retrieved the folder.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder.

coworker_config

The configuration parameters of the worker that retrieved the folder, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder.

coplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder.

coplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder.

coplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder.

coplugin_version

The version number of the plugin used by the worker that retrieved the folder.

coplugin_params

The parameters of the plugin used by the worker that retrieved the folder.

cofolder_class_parameters

The class parameters applied to the Cloud Observer folder and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

cofolder_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **cofolder_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

cofolder_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:

<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_folder_count — Count the number of Cloud Observer folders

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Chapter 50. Cloud Observer Instance

Name

co_instance_list — List the Cloud Observer instances

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

cofolder_id

The database identifier (ID) of the folder the object belongs to.

cofolder_name

The name of the folder the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder the object belongs to.

cofolder_status

The status of the folder the object belongs to.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coworker_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder the object belongs to was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder the object belongs to.

coworker_config

The configuration parameters of the worker that retrieved the folder the object belongs to, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder the object belongs to.

coplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

coinstance_id

The database identifier (ID) of the instance.

coinstance_name

The name of the instance.

coinstance_cpu

The CPU of the instance.

coinstance_ram

The RAM of the instance, in megabytes.

coinstance_cloud_id

The cloud provider database identifier (ID) of the object retrieved as an instance.

coinstance_firstseen

The first time the instance was seen, in decimal UNIX date format.

coinstance_updatetime

The last time any data regarding the instance was received, in decimal UNIX date format.

coinstance_laststatustime

The last time the status of the instance has been refreshed, in decimal UNIX date format.

coinstance_statusperiod

The instance status duration, in seconds.

coinstance_status

The status of the instance.

conetwork_id_list

The list of database identifiers (ID) of each of the attached networks of the instance.

conetwork_name_list

The list of names of each of the attached networks of the instance.

coinstance_class_parameters

The class parameters applied to the instance and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

coinstance_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **coinstance_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

coinstance_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_instance_info — Display the properties of a Cloud Observer instance

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

coinstance_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

coinstance_id

The database identifier (ID) of the instance, a unique numeric key value automatically incremented when the instance is retrieved. Use the ID to specify the instance of your choice.

Output Parameters

cofolder_id

The database identifier (ID) of the folder the object belongs to.

cofolder_name

The name of the folder the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder the object belongs to.

cofolder_status

The status of the folder the object belongs to.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coworker_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder the object belongs to was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder the object belongs to.

coworker_config

The configuration parameters of the worker that retrieved the folder the object belongs to, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder the object belongs to.

cplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

coinstance_id

The database identifier (ID) of the instance.

coinstance_name

The name of the instance.

coinstance_cpu

The CPU of the instance.

coinstance_ram

The RAM of the instance, in megabytes.

coinstance_cloud_id

The cloud provider database identifier (ID) of the object retrieved as an instance.

coinstance_firstseen

The first time the instance was seen, in decimal UNIX date format.

coinstance_updatetime

The last time any data regarding the instance was received, in decimal UNIX date format.

coinstance_laststatustime

The last time the status of the instance has been refreshed, in decimal UNIX date format.

coinstance_statusperiod

The instance status duration, in seconds.

coinstance_status

The status of the instance.

conetwork_id_list

The list of database identifiers (ID) of each of the attached networks of the instance.

conetwork_name_list

The list of names of each of the attached networks of the instance.

coinstance_class_parameters

The class parameters applied to the instance and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

coinstance_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by

coinstance_class_parameters: <class-parameter1>=<inheritance>, <propagation>&<class-parameter2>=<inheritance>&.... .

coinstance_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:

<class-parameter1>=real_<container-type>, <container-ID>&<class-parameter2>=real_<container-type>, <container-ID>&.... .

Name

co_instance_count — Count the number of Cloud Observer instances

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

co_instance_network_list — List the attached networks

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

cofolder_id

The database identifier (ID) of the folder the object belongs to.

cofolder_name

The name of the folder the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder the object belongs to.

cofolder_status

The status of the folder the object belongs to.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coworker_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder the object belongs to was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder the object belongs to.

coworker_config

The configuration parameters of the worker that retrieved the folder the object belongs to, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder the object belongs to.

coplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

coinstance_id

The database identifier (ID) of the instance linked to the attached network.

conetworkinstance_id

The database identifier (ID) of the link between the network and the instance.

coinstancenetwork_id

The database identifier (ID) of the link between the network and the instance.

coinstance_name

The name of the instance linked to the attached network.

coinstance_cpu

The CPU of the instance linked to the attached network.

coinstance_ram

The RAM of the instance linked to the attached network, in megabytes.

coinstance_firstseen

The first time the instance linked to the attached network was seen, in decimal UNIX date format.

coinstance_updatetime

The last time any data regarding the instance linked to the attached network was received, in decimal UNIX date format.

coinstance_laststatustime

The last time the status of the instance linked to the attached network has been refreshed, in decimal UNIX date format.

coinstance_statusperiod

The instance linked to the attached network status duration, in seconds.

coinstance_status

The status of the instance linked to the attached network.

conetwork_id

The database identifier (ID) of the attached network.

conetwork_name

The name of the attached network.

conetwork_cloud_id

The cloud provider database identifier (ID) of the object retrieved as an attached network.

conetwork_firstseen

The first time the attached network was seen, in decimal UNIX date format.

conetwork_updatetime

The last time any data regarding the attached network was received, in decimal UNIX date format.

conetwork_laststatustime

The last time the status of the attached network has been refreshed, in decimal UNIX date format.

conetwork_statusperiod

The attached network status duration, in seconds.

conetwork_status

The status of the attached network.

conetwork_class_parameters

The class parameters applied to the attached network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

conetwork_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **conetwork_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

conetwork_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_instance_network_info — Display the properties of an attached network

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

coinstancenetwork_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

coinstancenetwork_id

The database identifier (ID) of the link between the network and the instance, a unique numeric key value automatically incremented when the link is retrieved. Use the ID to specify the attached network of your choice.

Output Parameters

cofolder_id

The database identifier (ID) of the folder the object belongs to.

cofolder_name

The name of the folder the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder the object belongs to.

cofolder_status

The status of the folder the object belongs to.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coworker_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder the object belongs to was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder the object belongs to.

coworker_config

The configuration parameters of the worker that retrieved the folder the object belongs to, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder the object belongs to.

cplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

coinstance_id

The database identifier (ID) of the instance linked to the attached network.

conetworkinstance_id

The database identifier (ID) of the link between the network and the instance.

coinstancenetwork_id

The database identifier (ID) of the link between the network and the instance.

coinstance_name

The name of the instance linked to the attached network.

coinstance_cpu

The CPU of the instance linked to the attached network.

coinstance_ram

The RAM of the instance linked to the attached network, in megabytes.

coinstance_firstseen

The first time the instance linked to the attached network was seen, in decimal UNIX date format.

coinstance_updatetime

The last time any data regarding the instance linked to the attached network was received, in decimal UNIX date format.

coinstance_laststatustime

The last time the status of the instance linked to the attached network has been refreshed, in decimal UNIX date format.

coinstance_statusperiod

The instance linked to the attached network status duration, in seconds.

coinstance_status

The status of the instance linked to the attached network.

conetwork_id

The database identifier (ID) of the attached network.

conetwork_name

The name of the attached network.

conetwork_cloud_id

The cloud provider database identifier (ID) of the object retrieved as an attached network.

conetwork_firstseen

The first time the attached network was seen, in decimal UNIX date format.

conetwork_updatetime

The last time any data regarding the attached network was received, in decimal UNIX date format.

conetwork_laststatustime

The last time the status of the attached network has been refreshed, in decimal UNIX date format.

conetwork_statusperiod

The attached network status duration, in seconds.

conetwork_status

The status of the attached network.

conetwork_class_parameters

The class parameters applied to the attached network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

conetwork_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **conetwork_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

conetwork_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_instance_network_count — Count the number of attached networks

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Chapter 51. Cloud Observer Network

Name

co_network_list — List the Cloud Observer networks

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

cofolder_id

The database identifier (ID) of the folder the object belongs to.

cofolder_name

The name of the folder the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder the object belongs to.

cofolder_status

The status of the folder the object belongs to.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coworker_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder the object belongs to was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder the object belongs to.

coworker_config

The configuration parameters of the worker that retrieved the folder the object belongs to, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder the object belongs to.

coplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

conetwork_id

The database identifier (ID) of the network.

conetwork_name

The name of the network.

conetwork_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a network.

conetwork_firstseen

The first time the network was seen, in decimal UNIX date format.

conetwork_updatetime

The last time any data regarding the network was received, in decimal UNIX date format.

conetwork_laststatustime

The last time the status of the network has been refreshed, in decimal UNIX date format.

conetwork_statusperiod

The network status duration, in seconds.

conetwork_status

The status of the network.

conetwork_class_parameters

The class parameters applied to the network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

conetwork_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **conetwork_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

conetwork_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_network_info — Display the properties of a Cloud Observer network

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

conetwork_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

conetwork_id

The database identifier (ID) of the network, a unique numeric key value automatically incremented when the network is retrieved. Use the ID to specify the network of your choice.

Output Parameters

cofolder_id

The database identifier (ID) of the folder the object belongs to.

cofolder_name

The name of the folder the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder the object belongs to.

cofolder_status

The status of the folder the object belongs to.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coworker_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder the object belongs to was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder the object belongs to.

coworker_config

The configuration parameters of the worker that retrieved the folder the object belongs to, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder the object belongs to.

coplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

conetwork_id

The database identifier (ID) of the network.

conetwork_name

The name of the network.

conetwork_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a network.

conetwork_firstseen

The first time the network was seen, in decimal UNIX date format.

conetwork_updatetime

The last time any data regarding the network was received, in decimal UNIX date format.

conetwork_laststatustime

The last time the status of the network has been refreshed, in decimal UNIX date format.

conetwork_statusperiod

The network status duration, in seconds.

conetwork_status

The status of the network.

conetwork_class_parameters

The class parameters applied to the network and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

conetwork_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **conetwork_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....

conetwork_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_network_count — Count the number of Cloud Observer networks

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

co_network_instance_list — List the attached instances

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

cofolder_id

The database identifier (ID) of the folder the object belongs to.

cofolder_name

The name of the folder the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder the object belongs to.

cofolder_status

The status of the folder the object belongs to.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coworker_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder the object belongs to was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder the object belongs to.

coworker_config

The configuration parameters of the worker that retrieved the folder the object belongs to, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder the object belongs to.

coplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

coinstance_id

The database identifier (ID) of the attached instance.

conetworkinstance_id

The database identifier (ID) of the link between the network and the instance.

coinstancenetwork_id

The database identifier (ID) of the link between the network and the instance.

coinstance_name

The name of the attached instance.

coinstance_cpu

The CPU of the attached instance.

coinstance_ram

The RAM of the attached instance, in megabytes.

coinstance_firstseen

The first time the attached instance was seen, in decimal UNIX date format.

coinstance_updatetime

The last time any data regarding the attached instance was received, in decimal UNIX date format.

coinstance_laststatustime

The last time the status of the attached instance has been refreshed, in decimal UNIX date format.

coinstance_statusperiod

The attached instance status duration, in seconds.

coinstance_status

The status of the attached instance.

conetwork_id

The database identifier (ID) of the network linked to the attached instance.

conetwork_name

The name of the network linked to the attached instance.

conetwork_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a network linked to the attached instance.

conetwork_firstseen

The first time the network linked to the attached instance was seen, in decimal UNIX date format.

conetwork_updatetime

The last time any data regarding the network linked to the attached instance was received, in decimal UNIX date format.

conetwork_laststatustime

The last time the status of the network linked to the attached instance has been refreshed, in decimal UNIX date format.

conetwork_statusperiod

The network linked to the attached instance status duration, in seconds.

conetwork_status

The status of the network linked to the attached instance.

conetwork_class_parameters

The class parameters applied to the network linked to the attached instance and their value:
<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

conetwork_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
conetwork_class_parameters: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .*

conetwork_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:
<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_network_instance_info — Display the properties of an attached instance

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

conetworkinstance_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

conetworkinstance_id

The database identifier (ID) of the link between the network and the instance, a unique numeric key value automatically incremented when the link is retrieved. Use the ID to specify the attached instance of your choice.

Output Parameters

cofolder_id

The database identifier (ID) of the folder the object belongs to.

cofolder_name

The name of the folder the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder the object belongs to.

cofolder_status

The status of the folder the object belongs to.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coworker_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder the object belongs to was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder the object belongs to.

coworker_config

The configuration parameters of the worker that retrieved the folder the object belongs to, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder the object belongs to.

cplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

coinstance_id

The database identifier (ID) of the attached instance.

conetworkinstance_id

The database identifier (ID) of the link between the network and the instance.

coinstancenetwork_id

The database identifier (ID) of the link between the network and the instance.

coinstance_name

The name of the attached instance.

coinstance_cpu

The CPU of the attached instance.

coinstance_ram

The RAM of the attached instance, in megabytes.

coinstance_firstseen

The first time the attached instance was seen, in decimal UNIX date format.

coinstance_updatetime

The last time any data regarding the attached instance was received, in decimal UNIX date format.

coinstance_laststatustime

The last time the status of the attached instance has been refreshed, in decimal UNIX date format.

coinstance_statusperiod

The attached instance status duration, in seconds.

coinstance_status

The status of the attached instance.

conetwork_id

The database identifier (ID) of the network linked to the attached instance.

conetwork_name

The name of the network linked to the attached instance.

conetwork_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a network linked to the attached instance.

conetwork_firstseen

The first time the network linked to the attached instance was seen, in decimal UNIX date format.

conetwork_updatetime

The last time any data regarding the network linked to the attached instance was received, in decimal UNIX date format.

conetwork_laststatustime

The last time the status of the network linked to the attached instance has been refreshed, in decimal UNIX date format.

conetwork_statusperiod

The network linked to the attached instance status duration, in seconds.

conetwork_status

The status of the network linked to the attached instance.

conetwork_class_parameters

The class parameters applied to the network linked to the attached instance and their value:

<class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

conetwork_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by

conetwork_class_parameters: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

conetwork_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:

<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_network_instance_count — Count the number of attached instances

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Chapter 52. Cloud Observer IP Address

Name

co_ip_list — List the Cloud Observer IP addresses

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

cofolder_id

The database identifier (ID) of the folder the object belongs to.

cofolder_name

The name of the folder the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder the object belongs to.

cofolder_status

The status of the folder the object belongs to.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coworker_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder the object belongs to was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder the object belongs to.

coworker_config

The configuration parameters of the worker that retrieved the folder the object belongs to, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder the object belongs to.

coplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

coinstance_id

The database identifier (ID) of the instance the object belongs to.

coinstance_name

The name of the instance the object belongs to.

coinstance_cpu

The CPU of the instance the object belongs to.

coinstance_ram

The RAM of the instance the object belongs to, in megabytes.

conetwork_id

The database identifier (ID) of the network the object belongs to.

conetwork_name

The name of the network the object belongs to.

conetworkinstance_id

The database identifier (ID) of the link between the network and the instance the object belongs to.

coinstancenetwork_id

The database identifier (ID) of the link between the network and the instance the object belongs to.

coip_id

The database identifier (ID) of the IP address.

coip_addr

The IPv4 or IPv6 address in human readable format.

coip_addr4

The IPv4 address in hexadecimal format.

coip_addr6

The IPv6 address in hexadecimal format.

coip_mac

The MAC address of the IP address.

coip_public

The type of the IP address, either Public (1) or Private (0).

coip_cloud_id

The cloud provider database identifier (ID) of the IP address.

coip_firstseen

The first time the IP address was seen, in decimal UNIX date format.

coip_updatetime

The last time any data regarding the IP address was received, in decimal UNIX date format.

coip_class_parameters

The class parameters applied to the IP address and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

coip_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **coip_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

coip_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_ip_info — Display the properties of a Cloud Observer IP address

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

coip_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

coip_id

The database identifier (ID) of the IP address, a unique numeric key value automatically incremented when the IP address is retrieved. Use the ID to specify the IP address of your choice.

Output Parameters

cofolder_id

The database identifier (ID) of the folder the object belongs to.

cofolder_name

The name of the folder the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

cofolder_cloud_id

The cloud provider database identifier (ID) of the object retrieved as a folder the object belongs to.

cofolder_status

The status of the folder the object belongs to.

coworker_uuid

The universally unique identifier (UUID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coworker_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coworker_updatetime

The last time any data regarding the worker that retrieved the folder the object belongs to was received, in decimal UNIX date format.

coworker_refresh_parameters

The scheduled database refresh frequency configured on the worker that retrieved the folder the object belongs to.

coworker_config

The configuration parameters of the worker that retrieved the folder the object belongs to, in JSON format.

coworker_status

The operational status of the worker that retrieved the folder the object belongs to.

cplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

cplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

coinstance_id

The database identifier (ID) of the instance the object belongs to.

coinstance_name

The name of the instance the object belongs to.

coinstance_cpu

The CPU of the instance the object belongs to.

coinstance_ram

The RAM of the instance the object belongs to, in megabytes.

conetwork_id

The database identifier (ID) of the network the object belongs to.

conetwork_name

The name of the network the object belongs to.

conetworkinstance_id

The database identifier (ID) of the link between the network and the instance the object belongs to.

coinstancenetwork_id

The database identifier (ID) of the link between the network and the instance the object belongs to.

coip_id

The database identifier (ID) of the IP address.

coip_addr

The IPv4 or IPv6 address in human readable format.

coip_addr4

The IPv4 address in hexadecimal format.

coip_addr6

The IPv6 address in hexadecimal format.

coip_mac

The MAC address of the IP address.

coip_public

The type of the IP address, either Public (1) or Private (0).

coip_cloud_id

The cloud provider database identifier (ID) of the IP address.

coip_firstseen

The first time the IP address was seen, in decimal UNIX date format.

coip_updatetime

The last time any data regarding the IP address was received, in decimal UNIX date format.

coip_class_parameters

The class parameters applied to the IP address and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

coip_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **coip_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

coip_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

Name

co_ip_count — Count the number of Cloud Observer IP addresses

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

co_ip_log_list — List the Cloud Observer addresses moved to the *IP address history*

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

coip_log_id

The database identifier (ID) of the IP address moved to the *IP address history*.

coip_log_addr

The IPv4 or IPv6 address of the address moved to the *IP address history*, in human readable format.

coip_log_addr4

The IPv4 address in hexadecimal format of the address moved to the *IP address history*.

coip_log_addr6

The IPv6 address in hexadecimal format of the address moved to the *IP address history*.

coip_log_mac

The MAC address of the IP address moved to the *IP address history*.

coip_log_public

The type of the IP address moved to the *IP address history*, either Public (1) or Private (0).

coip_log_firstseen

The first time the IP address moved to the *IP address history* was seen, in decimal UNIX date format.

coip_log_updatetime

The last time any data regarding the IP address moved to the *IP address history* was received, in decimal UNIX date format.

cofolder_name

The name of the folder the object belongs to.

coinstance_name

The name of the instance the object belongs to.

conetwork_name

The name of the network the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

copugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

copugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

coip_log_class_parameters

The class parameters applied to the IP address moved to the IP address history and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

coip_log_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **coip_log_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

coip_log_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_ip_log_info — Display the properties of a Cloud Observer address moved to the *IP address history*

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

coip_log_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

coip_log_id

The database identifier (ID) of the IP address moved to the *IP address history*, a unique numeric key value automatically incremented when the IP address is moved. Use the ID to specify the IP address of your choice.

Output Parameters

coip_log_id

The database identifier (ID) of the IP address moved to the *IP address history*.

coip_log_addr

The IPv4 or IPv6 address of the address moved to the *IP address history*, in human readable format.

coip_log_addr4

The IPv4 address in hexadecimal format of the address moved to the *IP address history*.

coip_log_addr6

The IPv6 address in hexadecimal format of the address moved to the *IP address history*.

coip_log_mac

The MAC address of the IP address moved to the *IP address history*.

coip_log_public

The type of the IP address moved to the *IP address history*, either Public (1) or Private (0).

coip_log_firstseen

The first time the IP address moved to the *IP address history* was seen, in decimal UNIX date format.

coip_log_updatetime

The last time any data regarding the IP address moved to the *IP address history* was received, in decimal UNIX date format.

cofolder_name

The name of the folder the object belongs to.

coinstance_name

The name of the instance the object belongs to.

conetwork_name

The name of the network the object belongs to.

coworker_id

The database identifier (ID) of the worker that retrieved the folder the object belongs to.

coworker_name

The name of the worker that retrieved the folder the object belongs to.

coplugin_id

The database identifier (ID) of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_name

The shortname of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_label

The name of the cloud provider of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_version

The version number of the plugin used by the worker that retrieved the folder the object belongs to.

coplugin_params

The parameters of the plugin used by the worker that retrieved the folder the object belongs to.

coip_log_class_parameters

The class parameters applied to the IP address moved to the IP address history and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

coip_log_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **coip_log_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

coip_log_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

co_ip_log_count — Count the number of Cloud Observer addresses moved to the *IP address history*

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Part IX. NetChange Services

Table of Contents

| | |
|-------------------------------------|------|
| 53. Network Device | 943 |
| iplocator_netdev_add | 944 |
| iplnetdev_list | 948 |
| iplnetdev_info | 953 |
| iplnetdev_count | 957 |
| iplnetdev_groupby | 958 |
| iplnetdev_groupby_count | 960 |
| group_iplnetdev_add | 962 |
| group_iplnetdev_delete | 964 |
| iplocator_netdev_delete | 966 |
| 54. IPv4 Route | 968 |
| iplnetdevroute_list | 969 |
| iplnetdevroute_info | 974 |
| iplnetdevroute_count | 978 |
| iplnetdevroute_groupby | 979 |
| iplnetdevroute_groupby_count | 981 |
| 55. IPv6 Route | 983 |
| iplnetdevroute6_list | 984 |
| iplnetdevroute6_info | 989 |
| iplnetdevroute6_count | 993 |
| iplnetdevroute6_groupby | 994 |
| iplnetdevroute6_groupby_count | 996 |
| 56. NetChange VLAN | 998 |
| iplnetdevvlan_list | 999 |
| iplnetdevvlan_count | 1002 |
| iplnetdevvlan_groupby | 1003 |
| iplnetdevvlan_groupby_count | 1005 |
| 57. Port | 1007 |
| iplocator_port_add | 1008 |
| iplport_list | 1012 |
| iplport_info | 1018 |
| iplport_count | 1024 |
| 58. NetChange IPv4 Address | 1025 |
| iplnetdevaddr_list | 1026 |
| iplnetdevaddr_info | 1028 |
| iplnetdevaddr_count | 1030 |
| iplnetdevaddr_groupby | 1031 |
| iplnetdevaddr_groupby_count | 1033 |
| 59. NetChange IPv6 Address | 1035 |
| iplnetdevaddr6_list | 1036 |
| iplnetdevaddr6_info | 1038 |
| iplnetdevaddr6_count | 1040 |
| iplnetdevaddr6_groupby | 1041 |
| iplnetdevaddr6_groupby_count | 1043 |
| 60. Discovered Item | 1045 |
| ipldev_list | 1046 |
| ipldev_count | 1050 |
| ipldev_groupby | 1051 |
| ipldev_groupby_count | 1053 |
| ipldev_log_count | 1055 |
| ipldev_log_list | 1056 |

Chapter 53. Network Device

Name

iplocator_netdev_add — Edit a NetChange network device

Description

This service allows you to edit existing objects, specified using an existing identifier. A call can only edit one object.

The value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

Editing: (iplnetdev_id || hostaddr)

Input Parameters

iplnetdev_id

The database identifier (ID) of the NetChange network device, a unique numeric key value automatically incremented when you add a NetChange network device. Use the ID to specify the device of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the NetChange network device.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

iplnetdev_addr

Deprecated, replaced by **hostaddr**.

site_id

The database identifier (ID) of the space associated with the NetChange network device. Use the ID to specify the space of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

iplnetdev_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

iplnetdev_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplnetdev_dot1x

The 802.1X authentication status of the NetChange network device, either enabled 1 or disabled 2.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rancid_sync

A way to enable (1) or disable (0) the configuration versioning on the NetChange network device you are adding.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

rancid_id

The database identifier (ID) of the configuration you want to associate with the NetChange network device.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplnetdev_retrieve_routes

A way to determine if you retrieve the routes (1) of the network device or not (0) during the refresh operation. By default, it is enabled as disabling this option is only relevant for devices managing a very large number of routes.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplnetdev_refresh_parameters

A way to configure a scheduled refresh of the database of the NetChange network device you are editing. The refresh frequency must be specified following the format <minute>:<hour>:<date-of-the-month>:<month>:<day-of-the-week> where:

- <minute> accepts values between 00 and 59.
- <hour> accepts values between 00 and 12.
- <date-of-the-month> accepts values between 1 and 31.
- <month> accepts values between 1 and 12, where 1 is January, etc.
- <day-of-the-week> accepts values between 1 and 7, where 1 is Monday, etc.

All periods must be specified and separated by : . Each period also accepts a valid regex pattern for datetime. Note that . * indicates any value of a period and allows, for instance, to set a refresh once a year on December 21st at 0h00 with 00:00:21:12:. * .

| | | | |
|------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
|------|--------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

iplnetdev_refresh_rancid_parameters

A way to configure a scheduled refresh of the configuration versioning of the NetChange network device you are adding. The refresh frequency must be specified following the format <minute>:<hour>:<date-of-the-month>:<month>:<day-of-the-week> where:

- <minute> accepts values between 00 and 59.
- <hour> accepts values between 00 and 12.
- <date-of-the-month> accepts values between 1 and 31.
- <month> accepts values between 1 and 12, where 1 is January, etc.
- <day-of-the-week> accepts values between 1 and 7, where 1 is Monday, etc.

All periods must be specified and separated by : . Each period also accepts a valid regex pattern for datetime. Note that .* indicates any value of a period and allows, for instance, to set a refresh once a week on Monday at 2h30 with 30:02:.*.*:1.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplnetdev_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>_class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.

- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

iplnetdev_list — List the NetChange network devices

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

site_name

The name of the space associated with the NetChange network device.

site_id

The database identifier (ID) of the space associated with the NetChange network device.

iplnetdev_ip_addr

The IPv4 address of the NetChange network device.

iplnetdev_ip6_addr

The IPv6 address of the NetChange network device.

iplnetdev_hostaddr

The human readable version of the parameter **iplnetdev_ip_addr**.

iplnetdev_mac

The MAC address of the NetChange network device.

iplnetdev_stack_id

The stack identifier (ID) of the NetChange network device.

iplnetdev_nb_stack

The total number of stacks of the NetChange network device.

iplnetdev_name

The name of the NetChange network device.

iplnetdev_synching

The synchronization status of the NetChange network device. 1 indicates that the device is currently being synchronized.

iplnetdev_uptime

The uptime of the NetChange network device, in seconds.

iplnetdev_sysname

The name associated with the NetChange network device for SNMP monitoring.

iplnetdev_syscontact

The contact associated with the NetChange network device for SNMP monitoring.

iplnetdev_type

The product name of the NetChange network device.

iplnetdev_vendor

The vendor name of the NetChange network device.

iplnetdev_descr

The description and operating system of the NetChange network device.

iplnetdev_vlanid

Internal use. Not documented.

iplnetdev_updatetime

The last time the NetChange network device has been refreshed, in decimal UNIX date format.

iplnetdev_analysistime

The time that was required to refresh the NetChange network device, in seconds.

iplnetdev_status

The status of the NetChange network device, either *OK* (1), in *timeout* (2) or *misconfigured* (3).

iplnetdev_nbports

The total number of ports the NetChange network device contains.

iplnetdev_nbusedports

The number of ports on the NetChange network device that are currently active.

iplnetdev_sysoid

The OID associated with the NetChange network device for SNMP monitoring.

iplnetdev_finished

Internal use. Not documented.

iplnetdev_refresh_parameters

The scheduled database refresh frequency configured of the NetChange network device.

iplnetdev_refresh_rancid_parameters

The scheduled configuration versioning refresh frequency of the NetChange network device.

iplnetdev_retrieve_routes

A way to determine if you retrieve the routes (1) of the network device or not (0) during the refresh operation. By default, it is enabled as disabling this option is only relevant for devices managing a very large number of routes.

iplnetdev_serial

The serial number of the NetChange network device.

iplnetdev_slot_serial

The slot number and slot serial number of used slots on the NetChange network device, as follows: <slot-number>:<slot-serial-number>. This parameter only provides information for used slots, empty slots are not listed.

iplnetdev_syslocation

The physical location associated with the NetChange network device for SNMP monitoring.

iplnetdev_class_name

The name of the class applied to the NetChange network device, it can be preceded by the class directory.

snmp_id

Internal use. Not documented.

iplnetdev_version

The version details of the NetChange network device.

iplnetdev_id

The database identifier (ID) of the NetChange network device.

iplnetdev_nbfreports

The number of ports on the NetChange network device that are not currently active.

iplnetdev_percusedports

The percentage of ports on the NetChange network device that are currently active.

iplnetdev_ip_addr_list

The list of IPv4 and/or IPv6 addresses configured for the interfaces of the NetChange network device. The addresses are returned in hexadecimal format as follows: <ip_address_1>,<ip_address_2>,etc. For more details, refer to the chapters [NetChange IPv4 Address](#) and [NetChange IPv6 Address](#).

supported_mibs

The list of MIBs supported by the NetChange network device.

iplnetdev_dot1x

The 802.1X authentication status of the NetChange network device, either *unsupported* (0), *enabled* (1) or *disabled* (2).

iplnetdev_port_security

The port-security status of the NetChange network device, either *unsupported* (0), *enabled* (1) or *disabled* (2).

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

iplnetdev_snmp_profile

The SNMP version used on the NetChange network device.

iplnetdev_rancid_id

The database identifier (ID) of the configuration associated with the NetChange network device.

iplnetdev_rancid_profile

The connection profile currently associated with the NetChange network device.

iplnetdev_rancid_revision

The revision number of the configuration associated with the NetChange network device.

iplnetdev_rancid_revision_id

The database identifier (ID) of the revision of the configuration associated with the NetChange network device.

iplnetdev_rancid_time

The time of the last editing of the NetChange network device configuration, in decimal UNIX date format.

iplnetdev_rancid_first_time

The time of creation of the NetChange network device configuration, in decimal UNIX date format.

iplnetdev_rancid_nochange_time

The time during which the NetChange network device configuration remained unchanged before the last configuration check, in seconds.

iplnetdev_rancid_check_time

The date of the last configuration check, in timestamp format.

iplnetdev_rancid_sync

The configuration versioning status of the NetChange network device, either *unsupported* (-1), *enabled* (0), *disabled* (1) or *error* (2).

iplnetdev_class_parameters

The class parameters applied to the NetChange network device and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

iplnetdev_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **iplnetdev_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

iplnetdev_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

iplnetdev_info — Display the properties of a NetChange network device

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

iplnetdev_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

iplnetdev_id

The database identifier (ID) of the NetChange network device, a unique numeric key value automatically incremented when you add a NetChange network device. Use the ID to specify the device of your choice.

Output Parameters

site_name

The name of the space associated with the NetChange network device.

site_id

The database identifier (ID) of the space associated with the NetChange network device.

iplnetdev_ip_addr

The IPv4 address of the NetChange network device.

iplnetdev_ip6_addr

The IPv6 address of the NetChange network device.

iplnetdev_hostaddr

The human readable version of the parameter `iplnetdev_ip_addr`.

iplnetdev_mac

The MAC address of the NetChange network device.

iplnetdev_stack_id

The stack identifier (ID) of the NetChange network device.

iplnetdev_nb_stack

The total number of stacks of the NetChange network device.

iplnetdev_name

The name of the NetChange network device.

iplnetdev_synching

The synchronization status of the NetChange network device. 1 indicates that the device is currently being synchronized.

iplnetdev_uptime

The uptime of the NetChange network device, in seconds.

iplnetdev_sysname

The name associated with the NetChange network device for SNMP monitoring.

iplnetdev_syscontact

The contact associated with the NetChange network device for SNMP monitoring.

iplnetdev_type

The product name of the NetChange network device.

iplnetdev_vendor

The vendor name of the NetChange network device.

iplnetdev_descr

The description and operating system of the NetChange network device.

iplnetdev_vlanid

Internal use. Not documented.

iplnetdev_updatetime

The last time the NetChange network device has been refreshed, in decimal UNIX date format.

iplnetdev_analysistime

The time that was required to refresh the NetChange network device, in seconds.

iplnetdev_status

The status of the NetChange network device, either *OK* (1), in *timeout* (2) or *misconfigured* (3).

iplnetdev_nbports

The total number of ports the NetChange network device contains.

iplnetdev_nbusedports

The number of ports on the NetChange network device that are currently active.

iplnetdev_sysoid

The OID associated with the NetChange network device for SNMP monitoring.

iplnetdev_finished

Internal use. Not documented.

iplnetdev_refresh_parameters

The scheduled database refresh frequency configured of the NetChange network device.

iplnetdev_refresh_rancid_parameters

The scheduled configuration versioning refresh frequency of the NetChange network device.

iplnetdev_retrieve_routes

A way to determine if you retrieve the routes (1) of the network device or not (0) during the refresh operation. By default, it is enabled as disabling this option is only relevant for devices managing a very large number of routes.

iplnetdev_serial

The serial number of the NetChange network device.

iplnetdev_slot_serial

The slot number and slot serial number of used slots on the NetChange network device, as follows: <slot-number>:<slot-serial-number>. This parameter only provides information for used slots, empty slots are not listed.

iplnetdev_syslocation

The physical location associated with the NetChange network device for SNMP monitoring.

iplnetdev_class_name

The name of the class applied to the NetChange network device, it can be preceded by the class directory.

snmp_id

Internal use. Not documented.

iplnetdev_version

The version details of the NetChange network device.

iplnetdev_id

The database identifier (ID) of the NetChange network device.

iplnetdev_nbfreeports

The number of ports on the NetChange network device that are not currently active.

iplnetdev_percusedports

The percentage of ports on the NetChange network device that are currently active.

iplnetdev_ip_addr_list

The list of IPv4 and/or IPv6 addresses configured for the interfaces of the NetChange network device. The addresses are returned in hexadecimal format as follows: <ip_address_1>,<ip_address_2>,etc. For more details, refer to the chapters [NetChange IPv4 Address](#) and [NetChange IPv6 Address](#).

supported_mibs

The list of MIBs supported by the NetChange network device.

iplnetdev_dot1x

The 802.1X authentication status of the NetChange network device, either *unsupported* (0), *enabled* (1) or *disabled* (2).

iplnetdev_port_security

The port-security status of the NetChange network device, either *unsupported* (0), *enabled* (1) or *disabled* (2).

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

iplnetdev_snmp_profile

The SNMP version used on the NetChange network device.

iplnetdev_rancid_id

The database identifier (ID) of the configuration associated with the NetChange network device.

iplnetdev_rancid_profile

The connection profile currently associated with the NetChange network device.

iplnetdev_rancid_revision

The revision number of the configuration associated with the NetChange network device.

iplnetdev_rancid_revision_id

The database identifier (ID) of the revision of the configuration associated with the NetChange network device.

iplnetdev_rancid_time

The time of the last editing of the NetChange network device configuration, in decimal UNIX date format.

iplnetdev_rancid_first_time

The time of creation of the NetChange network device configuration, in decimal UNIX date format.

iplnetdev_rancid_nochange_time

The time during which the NetChange network device configuration remained unchanged before the last configuration check, in seconds.

iplnetdev_rancid_check_time

The date of the last configuration check, in timestamp format.

iplnetdev_rancid_sync

The configuration versioning status of the NetChange network device, either *unsupported* (-1), *enabled* (0), *disabled* (1) or *error* (2).

iplnetdev_class_parameters

The class parameters applied to the NetChange network device and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

iplnetdev_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **iplnetdev_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

iplnetdev_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

iplnetdev_count — Count the number of NetChange network devices

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

iplnetdev_groupby — Group NetChange network devices by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

iplnetdev_groupby_count — Count the number of NetChange network devices grouped by parameter(s)

Description

This service allows you to display the total number of results of the service `*_groupby`.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement `GROUPBY`.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: `count`, `max`, `min`, `sum` or `avg`. The aggregation function syntax is the following: `SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)` where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement `SELECT` must also be specified in the statement `GROUPBY`.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement `SELECT` without aggregation function must be specified in the statement `GROUPBY`.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...* . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

group_iplnetdev_add — Add a NetChange network device to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (iplnetdev_id || (hostaddr && (site_id || site_name))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

iplnetdev_id

The database identifier (ID) of the NetChange network device, a unique numeric key value automatically incremented when you add a NetChange network device. Use the ID to specify the device of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the NetChange network device.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

iplnetdev_addr

Deprecated, replaced by **hostaddr**.

site_id

The database identifier (ID) of the space associated with the NetChange network device. Use the ID to specify the space of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space associated with the NetChange network device.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

group_ipNetdev_delete — Remove a NetChange network device from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (ipNetdev_id || (hostaddr && (site_id || site_name))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ipNetdev_id

The database identifier (ID) of the NetChange network device, a unique numeric key value automatically incremented when you add a NetChange network device. Use the ID to specify the device of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the NetChange network device.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

ipNetdev_addr

Deprecated, replaced by **hostaddr**.

site_id

The database identifier (ID) of the space associated with the NetChange network device. Use the ID to specify the space of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space associated with the NetChange network device.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

iplocator_netdev_delete — Delete a NetChange network device

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(iplnetdev_id || (hostaddr && (site_id || site_name)))

Input Parameters

iplnetdev_id

The database identifier (ID) of the NetChange network device, a unique numeric key value automatically incremented when you add a NetChange network device. Use the ID to specify the device of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostaddr

The IP address of the NetChange network device.

| Type | IPv4/Ipv6 address | Maximum length | N/A |
|---------------|-------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

iplnetdev_addr

Deprecated, replaced by **hostaddr**.

site_id

The database identifier (ID) of the space associated with the NetChange network device. Use the ID to specify the space of your choice.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

site_name

The name of the space associated with the NetChange network device.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 54. IPv4 Route

Name

iplnetdevroute_list — List the IPv4 routes

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

site_name

Internal use. Not documented.

site_id

Internal use. Not documented.

iplnetdev_ip_addr

The IPv4 address of the NetChange network device the object belongs to.

iplnetdev_ip6_addr

The IPv6 address of the NetChange network device the object belongs to.

iplnetdev_stack_id

The stack identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_nb_stack

The total number of stacks of the NetChange network device the object belongs to.

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplnetdev_synching

The synchronization status of the NetChange network device the object belongs to. 1 indicates that the device is currently being synchronized.

iplnetdev_uptime

The uptime of the NetChange network device the object belongs to, in seconds.

iplnetdev_sysname

The name associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_syscontact

The contact associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_type

The product name of the NetChange network device the object belongs to.

iplnetdev_vendor

The vendor name of the NetChange network device the object belongs to.

iplnetdev_descr

The description and operating system of the NetChange network device the object belongs to.

iplnetdev_vlanid

Internal use. Not documented.

iplnetdev_updatetime

The last time the NetChange network device the object belongs to has been refreshed, in decimal UNIX date format.

iplnetdev_analysistime

The time that was required to refresh the NetChange network device the object belongs to, in seconds.

iplnetdev_status

The status of the NetChange network device the object belongs to, either *OK* (1), in *timeout* (2) or *misconfigured* (3).

iplnetdev_nbports

The total number of ports the NetChange network device the object belongs to contains.

iplnetdev_nbusedports

The number of ports on the NetChange network device the object belongs to that are currently active.

iplnetdev_sysoid

The OID associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_finished

Internal use. Not documented.

iplnetdev_refresh_parameters

The scheduled database refresh frequency configured on the NetChange network device the object belongs to.

iplnetdev_refresh_rancid_parameters

The configuration versioning refresh frequency on the NetChange network device the object belongs to.

iplnetdev_serial

The serial number of the NetChange network device the object belongs to.

iplnetdev_syslocation

The physical location associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_class_name

The name of the class applied to the NetChange network device the object belongs to, it can be preceded by the class directory.

iplnetdev_class_parameters

The class parameters applied to the NetChange network device the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

snmp_id

Internal use. Not documented.

iplnetdev_version

The version details of the NetChange network device the object belongs to.

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_nbfreeports

The number of ports on the NetChange network device the object belongs to that are not currently active.

iplnetdev_percusedports

The percentage of ports on the NetChange network device the object belongs to that are currently active.

iplnetdevroute_id

The database identifier (ID) of the IPv4 route.

iplnetdevroute_ip_addr

The IPv4 address of the route.

iplnetdevroute_prefix

The prefix size of the IPv4 route.

iplnetdevroute_nexthop

The next router to reach the destination within the VRF.

iplnetdevroute_type

The type of route:

- *other* indicates that the routing is not working, we do not know why.
- *invalid* or *reject* indicates that the route is invalidated or discarding traffic.
- *local* or *direct* indicates that the route originates from a local interface.
- *remote* indicates that the route has a remote destination.
- *blackhole* indicates that the route is discarding traffic silently.

iplnetdevroute_proto

The protocol via which the route was found.

Table 54.1. *iplnetdevroute_proto* possible values

| Value | Description |
|------------|--|
| other | No protocol is specified. |
| local | Local interface. |
| netmgmt | Static route. |
| icmp | Result of ICMP Redirect. |
| egp | Exterior Gateway Protocol. |
| ggp | Gateway-Gateway Protocol. |
| hello | FuzzBall HelloSpeak |
| rip | Berkeley RIP or RIP-II. |
| is-is | Dual IS-IS. |
| es-is | ISO 9542. |
| ciscolgrp | Cisco IGRP.o |
| bbnSpfIgp | BBN SPF IGP. |
| ospf | Open Shortest Path First. |
| bgp | Border Gateway Protocol. |
| idpr | InterDomain Policy Routing. |
| ciscoEigrp | Cisco Enhanced Interior Gateway Routing Protocol. |
| dvmrp | Distance Vector Multicast Routing Protocol. |
| rpl | Routing Protocol for LLNs [RFC-ietf-roll-rpl-19]. |
| dhcp | Dynamic Host Configuration Protocol [RFC2132]. |
| ttdp | Train Topology Discovery Protocol [IEC 61375-2-5]. |

iplvrf_name

The name of the VRF, as set on the network device the object belongs to.

iplvrf_rd

The Route Distinguisher of **iplvrf_name**. A unique 64-bits identifier, that can be composed of IP addresses or AS Numbers, that differentiates every set of routes on each VRF.

iplnetdevip_ip_addr_list

The list of IPv4 and/or IPv6 addresses configured for the interfaces of the NetChange network device the object belongs to. The addresses are returned in hexadecimal format as follows: <ip_address_1>,<ip_address_2>,etc. For more details, refer to the chapters [NetChange IPv4 Address](#) and [NetChange IPv6 Address](#).

Name

iplnetdevroute_info — Display the properties of an IPv4 route

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

iplnetdevroute_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

iplnetdevroute_id

The database identifier (ID) of the IPv4 route, a unique numeric key value automatically incremented when the route was discovered on the device. Use it to identify the route of your choice.

Output Parameters

site_name

Internal use. Not documented.

site_id

Internal use. Not documented.

iplnetdev_ip_addr

The IPv4 address of the NetChange network device the object belongs to.

iplnetdev_ip6_addr

The IPv6 address of the NetChange network device the object belongs to.

iplnetdev_stack_id

The stack identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_nb_stack

The total number of stacks of the NetChange network device the object belongs to.

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplnetdev_synching

The synchronization status of the NetChange network device the object belongs to. 1 indicates that the device is currently being synchronized.

iplnetdev_uptime

The uptime of the NetChange network device the object belongs to, in seconds.

iplnetdev_sysname

The name associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_syscontact

The contact associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_type

The product name of the NetChange network device the object belongs to.

iplnetdev_vendor

The vendor name of the NetChange network device the object belongs to.

iplnetdev_descr

The description and operating system of the NetChange network device the object belongs to.

iplnetdev_vlanid

Internal use. Not documented.

iplnetdev_updatetime

The last time the NetChange network device the object belongs to has been refreshed, in decimal UNIX date format.

iplnetdev_analysistime

The time that was required to refresh the NetChange network device the object belongs to, in seconds.

iplnetdev_status

The status of the NetChange network device the object belongs to, either *OK* (1), in *timeout* (2) or *misconfigured* (3).

iplnetdev_nbports

The total number of ports the NetChange network device the object belongs to contains.

iplnetdev_nbusedports

The number of ports on the NetChange network device the object belongs to that are currently active.

iplnetdev_sysoid

The OID associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_finished

Internal use. Not documented.

iplnetdev_refresh_parameters

The scheduled database refresh frequency configured on the NetChange network device the object belongs to.

iplnetdev_refresh_rancid_parameters

The configuration versioning refresh frequency on the NetChange network device the object belongs to.

iplnetdev_serial

The serial number of the NetChange network device the object belongs to.

iplnetdev_syslocation

The physical location associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_class_name

The name of the class applied to the NetChange network device the object belongs to, it can be preceded by the class directory.

iplnetdev_class_parameters

The class parameters applied to the NetChange network device the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

snmp_id

Internal use. Not documented.

iplnetdev_version

The version details of the NetChange network device the object belongs to.

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_nbfreeports

The number of ports on the NetChange network device the object belongs to that are not currently active.

iplnetdev_percusedports

The percentage of ports on the NetChange network device the object belongs to that are currently active.

iplnetdevroute_id

The database identifier (ID) of the IPv4 route.

iplnetdevroute_ip_addr

The IPv4 address of the route.

iplnetdevroute_prefix

The prefix size of the IPv4 route.

iplnetdevroute_nexthop

The next router to reach the destination within the VRF.

iplnetdevroute_type

The type of route:

- *other* indicates that the routing is not working, we do not know why.
- *invalid* or *reject* indicates that the route is invalidated or discarding traffic.
- *local* or *direct* indicates that the route originates from a local interface.
- *remote* indicates that the route has a remote destination.
- *blackhole* indicates that the route is discarding traffic silently.

iplnetdevroute_proto

The protocol via which the route was found.

Table 54.2. iplnetdevroute_proto possible values

| Value | Description |
|------------|--|
| other | No protocol is specified. |
| local | Local interface. |
| netmgmt | Static route. |
| icmp | Result of ICMP Redirect. |
| egp | Exterior Gateway Protocol. |
| ggp | Gateway-Gateway Protocol. |
| hello | FuzzBall HelloSpeak |
| rip | Berkeley RIP or RIP-II. |
| is-is | Dual IS-IS. |
| es-is | ISO 9542. |
| ciscolgrp | Cisco IGRP.o |
| bbnSpflgp | BBN SPF IGP. |
| ospf | Open Shortest Path First. |
| bgp | Border Gateway Protocol. |
| idpr | InterDomain Policy Routing. |
| ciscoEigrp | Cisco Enhanced Interior Gateway Routing Protocol. |
| dvmrp | Distance Vector Multicast Routing Protocol. |
| rpl | Routing Protocol for LLNs [RFC-ietf-roll-rpl-19]. |
| dhcp | Dynamic Host Configuration Protocol [RFC2132]. |
| ttdp | Train Topology Discovery Protocol [IEC 61375-2-5]. |

iplvrf_name

The name of the VRF, as set on the network device the object belongs to.

iplvrf_rd

The Route Distinguisher of **iplvrf_name**. A unique 64-bits identifier, that can be composed of IP addresses or AS Numbers, that differentiates every set of routes on each VRF.

iplnetdevip_ip_addr_list

The list of IPv4 and/or IPv6 addresses configured for the interfaces of the NetChange network device the object belongs to. The addresses are returned in hexadecimal format as follows: <ip_address_1>,<ip_address_2>,etc. For more details, refer to the chapters [NetChange IPv4 Address](#) and [NetChange IPv6 Address](#).

Name

iplnetdevroute_count — Count the number of IPv4 routes

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

iplnetdevroute_groupby — Group IPv4 routes by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

iplnetdevroute_groupby_count — Count the number of IPv4 routes grouped by parameter(s)

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Chapter 55. IPv6 Route

Name

iplnetdevroute6_list — List the IPv6 routes

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

site_name

Internal use. Not documented.

site_id

Internal use. Not documented.

iplnetdev_ip_addr

The IPv4 address of the NetChange network device the object belongs to.

iplnetdev_ip6_addr

The IPv6 address of the NetChange network device the object belongs to.

iplnetdev_stack_id

The stack identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_nb_stack

The total number of stacks of the NetChange network device the object belongs to.

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplnetdev_synching

The synchronization status of the NetChange network device the object belongs to. 1 indicates that the device is currently being synchronized.

iplnetdev_uptime

The uptime of the NetChange network device the object belongs to, in seconds.

iplnetdev_sysname

The name associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_syscontact

The contact associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_type

The product name of the NetChange network device the object belongs to.

iplnetdev_vendor

The vendor name of the NetChange network device the object belongs to.

iplnetdev_descr

The description and operating system of the NetChange network device the object belongs to.

iplnetdev_vlanid

Internal use. Not documented.

iplnetdev_updatetime

The last time the NetChange network device the object belongs to has been refreshed, in decimal UNIX date format.

iplnetdev_analysistime

The time that was required to refresh the NetChange network device the object belongs to, in seconds.

iplnetdev_status

The status of the NetChange network device the object belongs to, either *OK* (1), in *timeout* (2) or *misconfigured* (3).

iplnetdev_nbports

The total number of ports the NetChange network device the object belongs to contains.

iplnetdev_nbusedports

The number of ports on the NetChange network device the object belongs to that are currently active.

iplnetdev_sysoid

The OID associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_finished

Internal use. Not documented.

iplnetdev_refresh_parameters

The scheduled database refresh frequency configured on the NetChange network device the object belongs to.

iplnetdev_refresh_rancid_parameters

The configuration versioning refresh frequency on the NetChange network device the object belongs to.

iplnetdev_serial

The serial number of the NetChange network device the object belongs to.

iplnetdev_syslocation

The physical location associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_class_name

The name of the class applied to the NetChange network device the object belongs to, it can be preceded by the class directory.

iplnetdev_class_parameters

The class parameters applied to the NetChange network device the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

snmp_id

Internal use. Not documented.

iplnetdev_version

The version details of the NetChange network device the object belongs to.

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_nbfreeports

The number of ports on the NetChange network device the object belongs to that are not currently active.

iplnetdev_percusedports

The percentage of ports on the NetChange network device the object belongs to that are currently active.

iplnetdevroute6_id

The database identifier (ID) of the IPv6 route.

iplnetdevroute6_ip6_addr

The IPv6 address of the route.

iplnetdevroute6_prefix

The prefix size of the IPv6 route.

iplnetdevroute6_nexthop

The next router to reach the destination within the VRF.

iplnetdevroute6_type

The type of route:

- *other* indicates that the routing is not working, we do not know why.
- *invalid* or *reject* indicates that the route is invalidated or discarding traffic.
- *local* or *direct* indicates that the route originates from a local interface.
- *remote* indicates that the route has a remote destination.
- *blackhole* indicates that the route is discarding traffic silently.

iplnetdevroute6_proto

The protocol via which the route was found.

Table 55.1. *iplnetdevroute_proto* possible values

| Value | Description |
|------------|--|
| other | No protocol is specified. |
| local | Local interface. |
| netmgmt | Static route. |
| icmp | Result of ICMP Redirect. |
| egp | Exterior Gateway Protocol. |
| ggp | Gateway-Gateway Protocol. |
| hello | FuzzBall HelloSpeak |
| rip | Berkeley RIP or RIP-II. |
| is-is | Dual IS-IS. |
| es-is | ISO 9542. |
| ciscolgrp | Cisco IGRP.o |
| bbnSpfIgp | BBN SPF IGP. |
| ospf | Open Shortest Path First. |
| bgp | Border Gateway Protocol. |
| idpr | InterDomain Policy Routing. |
| ciscoEigrp | Cisco Enhanced Interior Gateway Routing Protocol. |
| dvmrp | Distance Vector Multicast Routing Protocol. |
| rpl | Routing Protocol for LLNs [RFC-ietf-roll-rpl-19]. |
| dhcp | Dynamic Host Configuration Protocol [RFC2132]. |
| ttdp | Train Topology Discovery Protocol [IEC 61375-2-5]. |

iplvrf_name

The name of the VRF, as set on the network device the object belongs to.

iplvrf_rd

The Route Distinguisher of **iplvrf_name**. A unique 64-bits identifier, that can be composed of IP addresses or AS Numbers, that differentiates every set of routes on each VRF.

iplnetdevip_ip_addr_list

The list of IPv4 and/or IPv6 addresses configured for the interfaces of the NetChange network device the object belongs to. The addresses are returned in hexadecimal format as follows: <ip_address_1>,<ip_address_2>,etc. For more details, refer to the chapters [NetChange IPv4 Address](#) and [NetChange IPv6 Address](#).

Name

iplnetdevroute6_info — Display the properties of an IPv6 route

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

iplnetdevroute6_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

iplnetdevroute6_id

The database identifier (ID) of the IPv6 route, a unique numeric key value automatically incremented when the route was discovered on the device. Use it to identify the route of your choice.

Output Parameters

site_name

Internal use. Not documented.

site_id

Internal use. Not documented.

iplnetdev_ip_addr

The IPv4 address of the NetChange network device the object belongs to.

iplnetdev_ip6_addr

The IPv6 address of the NetChange network device the object belongs to.

iplnetdev_stack_id

The stack identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_nb_stack

The total number of stacks of the NetChange network device the object belongs to.

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplnetdev_synching

The synchronization status of the NetChange network device the object belongs to. 1 indicates that the device is currently being synchronized.

iplnetdev_uptime

The uptime of the NetChange network device the object belongs to, in seconds.

iplnetdev_sysname

The name associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_syscontact

The contact associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_type

The product name of the NetChange network device the object belongs to.

iplnetdev_vendor

The vendor name of the NetChange network device the object belongs to.

iplnetdev_descr

The description and operating system of the NetChange network device the object belongs to.

iplnetdev_vlanid

Internal use. Not documented.

iplnetdev_updatetime

The last time the NetChange network device the object belongs to has been refreshed, in decimal UNIX date format.

iplnetdev_analysistime

The time that was required to refresh the NetChange network device the object belongs to, in seconds.

iplnetdev_status

The status of the NetChange network device the object belongs to, either *OK* (1), in *timeout* (2) or *misconfigured* (3).

iplnetdev_nbports

The total number of ports the NetChange network device the object belongs to contains.

iplnetdev_nbusedports

The number of ports on the NetChange network device the object belongs to that are currently active.

iplnetdev_sysoid

The OID associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_finished

Internal use. Not documented.

iplnetdev_refresh_parameters

The scheduled database refresh frequency configured on the NetChange network device the object belongs to.

iplnetdev_refresh_rancid_parameters

The configuration versioning refresh frequency on the NetChange network device the object belongs to.

iplnetdev_serial

The serial number of the NetChange network device the object belongs to.

iplnetdev_syslocation

The physical location associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_class_name

The name of the class applied to the NetChange network device the object belongs to, it can be preceded by the class directory.

iplnetdev_class_parameters

The class parameters applied to the NetChange network device the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

snmp_id

Internal use. Not documented.

iplnetdev_version

The version details of the NetChange network device the object belongs to.

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_nbfreeports

The number of ports on the NetChange network device the object belongs to that are not currently active.

iplnetdev_percusedports

The percentage of ports on the NetChange network device the object belongs to that are currently active.

iplnetdevroute6_id

The database identifier (ID) of the IPv6 route.

iplnetdevroute6_ip6_addr

The IPv6 address of the route.

iplnetdevroute6_prefix

The prefix size of the IPv6 route.

iplnetdevroute6_nexthop

The next router to reach the destination within the VRF.

iplnetdevroute6_type

The type of route:

- *other* indicates that the routing is not working, we do not know why.
- *invalid* or *reject* indicates that the route is invalidated or discarding traffic.
- *local* or *direct* indicates that the route originates from a local interface.
- *remote* indicates that the route has a remote destination.
- *blackhole* indicates that the route is discarding traffic silently.

iplnetdevroute6_proto

The protocol via which the route was found.

Table 55.2. ipnetdevroute_proto possible values

| Value | Description |
|------------|--|
| other | No protocol is specified. |
| local | Local interface. |
| netmgmt | Static route. |
| icmp | Result of ICMP Redirect. |
| egp | Exterior Gateway Protocol. |
| ggp | Gateway-Gateway Protocol. |
| hello | FuzzBall HelloSpeak |
| rip | Berkeley RIP or RIP-II. |
| is-is | Dual IS-IS. |
| es-is | ISO 9542. |
| ciscolgrp | Cisco IGRP.o |
| bbnSpflgp | BBN SPF IGP. |
| ospf | Open Shortest Path First. |
| bgp | Border Gateway Protocol. |
| idpr | InterDomain Policy Routing. |
| ciscoEigrp | Cisco Enhanced Interior Gateway Routing Protocol. |
| dvmrp | Distance Vector Multicast Routing Protocol. |
| rpl | Routing Protocol for LLNs [RFC-ietf-roll-rpl-19]. |
| dhcp | Dynamic Host Configuration Protocol [RFC2132]. |
| ttdp | Train Topology Discovery Protocol [IEC 61375-2-5]. |

iplvrf_name

The name of the VRF, as set on the network device the object belongs to.

iplvrf_rd

The Route Distinguisher of **iplvrf_name**. A unique 64-bits identifier, that can be composed of IP addresses or AS Numbers, that differentiates every set of routes on each VRF.

ipnetdevip_ip_addr_list

The list of IPv4 and/or IPv6 addresses configured for the interfaces of the NetChange network device the object belongs to. The addresses are returned in hexadecimal format as follows: <ip_address_1>, <ip_address_2>, etc. For more details, refer to the chapters [NetChange IPv4 Address](#) and [NetChange IPv6 Address](#).

Name

iplnetdevroute6_count — Count the number of IPv6 routes

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

iplnetdevroute6_groupby — Group IPv6 routes by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: <parameter>='<value>'. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement **SELECT** is returned.

Name

iplnetdevroute6_groupby_count — Count the number of IPv6 routes grouped by parameter(s)

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Chapter 56. NetChange VLAN

Name

iplnetdevvlan_list — List the NetChange VLANs

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

iplnetdevvlan_id

The database identifier (ID) of the NetChange VLAN, a unique numeric key value automatically incremented when you add a NetChange VLAN.

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplnetdevvlan_number

The VLAN identifier (ID) of the NetChange VLAN.

iplnetdevvlan_name

The name of the NetChange VLAN.

iplnetdevvlan_status

The status of the NetChange VLAN, either *OK* (1), *inactive* (2) or *dynamic* (3).

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplnetdev_class_name

The name of the class applied to the NetChange network device the object belongs to, it can be preceded by the class directory.

iplnetdev_nbports

The total number of ports the NetChange network device the object belongs to contains.

iplnetdev_type

The product name of the NetChange network device the object belongs to.

iplnetdev_vendor

The vendor name of the NetChange network device the object belongs to.

iplnetdev_syscontact

The contact associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_serial

The serial number of the NetChange network device the object belongs to.

iplnetdev_status

The status of the NetChange network device the object belongs to, either *OK* (1), in *timeout* (2) or *misconfigured* (3).

iplnetdev_syslocation

The physical location associated with the NetChange network device the object belongs to for SNMP monitoring.

vlan_port_name_list

The name of the ports associated with the NetChange VLAN, as follows: <iplport_name>, <iplport_name>,

vlan_access_port_name_list

The list of ports associated with the NetChange VLAN set with the mode Access.

Name

iplnetdevvlan_count — Count the number of NetChange VLANs

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

iplnetdevvlan_groupby — Group NetChange VLANs by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

iplnetdevvlan_groupby_count — Count the number of NetChange VLANs grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Chapter 57. Port

Name

iplocator_port_add — Edit a port

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

Editing: iplport_id

Input Parameters**iplport_id**

The database identifier (ID) of the port, a unique numeric key value automatically incremented when you add a port. Use the ID to specify the port of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_ifoperstatus

The operational status of the port, you can set it to *enabled* or *disabled*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_description

The description of the port.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_ifcfgspeed

The configured speed of the NetChange port, in bits per seconds (bps). Set it to -1 to indicate that the speed is automatically configured (*auto*).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_cfg_duplex

The configured duplex mode of the NetChange port, either *automatic* (*auto*) *half-duplex* (*half*) or *full-duplex* (*full*).

| | | | |
|---------------|-----------------------------------|----------------|-----|
| Type | Fixed value: half full auto | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_dot1x

The 802.1X authentication status of the NetChange port, either *unsupported* (0), *enabled* (1) or *disabled* (2).

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_port_security

The port-security mode of the NetChange port, either *unsupported* (0), *disabled* (1), *FirstN* (2), *configureSpecific* (4), *8021xAuthorized* (5) or *LimitedContinuous* (6).

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_port_security_max_mac

The maximum number of MAC addresses allowed to access the port. The port-security mode must be enabled.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_port_security_action

The port-security action of the NetChange port. Possible values can be:

- For HP devices: *disable* (1), *sendTrap* (2) or *sendTrapAndDisablePort* (3).
- For Cisco devices: *shutdown* (4), *dropNotify* (5) or *drop* (6).
- For Juniper devices: *none* (1), *drop* (2), *alarm* (3) or *shutdown* (4).

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_vlan_tagging

The VLAN tagging status of the NetChange port. Set it to 1 to enable VLAN tagging.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_untagged_vlan

The list of the VLAN identifier (ID) of the untagged NetChange VLANs associated with the NetChange port, as follows: <vlan_id>, <vlan_id>... .

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_tagged_vlans

The list of the VLAN identifier (ID) of the tagged NetChange VLANs associated with the NetChange port, as follows: <vlan_id>, <vlan_id>... .

| | | | |
|----------------------|---|-----------------------|-----|
| Type | Regular expression: ^([0-9]+([,][0-9]+)*)?\$/ | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_interco_mode

The interconnection status of the NetChange port. 1 indicates that interconnection is enabled.

| | | | |
|---------------|--------------------------------|----------------|-----|
| Type | Fixed value: auto yes no | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>_class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... . If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.

- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

iplport_list — List the ports

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

port_vlan_number_list

The list of the VLAN identifier (ID) of all the NetChange VLANs associated with the NetChange port, as follows: <vlan_id>, <vlan_id>....

port_vlan_name_list

The list of the name of all the NetChange VLANs associated with the NetChange port, as follows: <vlan_id>, <vlan_id>....

tagged_vlan_list

The list of the VLAN identifier (ID) of the tagged NetChange VLANs associated with the NetChange port, as follows: <vlan_id>, <vlan_id>....

iplport_oper_dot1x

The operating 802.1X authentication status of the NetChange port, either *unsupported* (0), *enabled* (1) or *disabled* (2).

iplport_port_security

The port-security mode of the NetChange port, either *unsupported* (0), *disabled* (1), *FirstN* (2), *configureSpecific* (4), *8021xAuthorized* (5) or *LimitedContinuous* (6).

iplport_dot1x

The configured 802.1X authentication status of the NetChange port, either *unsupported* (0), *enabled* (1) or *disabled* (2).

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplport_ifnumber

The interface number of the NetChange port, as follows: <slot_number>. <port_number>, or <port_number> only when the slot number is 0.

iplport_name

The name of the port.

iplport_description

The description of the port.

iplport_ifname

The internal monitoring name of the port.

iplport_iftype

The type of the NetChange port.

iplport_ifdescr

The internal monitoring description of the NetChange port.

iplport_ifaddress

Internal use. Not documented.

iplport_ifvlan

Internal use. Not documented.

iplport_untagged_vlan

The list of the VLAN identifier (ID) of the untagged NetChange VLANs associated with the NetChange port, as follows: <vlan_id>, <vlan_id>... .

iplport_vlan_tagging

The VLAN tagging status of the NetChange port, either *trunk* (1), *access* (2), *auto* (4), *mixed* (6) or *tagged* (7).

iplport_trunk_status

The VLAN trunking status of the NetChange port, either *N/A* (0), *Trunk/Tagged* (1), or *Access/Untagged* (2).

iplport_ifspeed_max

The maximum speed of the NetChange port, in bits per seconds (bps). 0 indicates that the port is not active.

iplport_ifvendor

The vendor name of the NetChange port.

iplport_ifoperspeed

The operating speed of the NetChange port, in bits per seconds (bps). 0 indicates that the port is not active.

iplport_ifcfgspeed

The configured speed of the NetChange port, in bits per seconds (bps). -1 indicates that the speed is automatically configured (*auto*).

iplport_ifoperstatus

The operational status of the port:

Table 57.1. iplport_ifoperstatus possible values

| Status | Description |
|----------------|---|
| active | The port is active, or up. |
| inactive | The port is inactive, or down. |
| testing | The port is up but no operational packets can be passed. |
| lowerLayerDown | |
| notPresent | The port is inactive. These statuses are very rare. For more details, refer to the description of the MIB <i>IF-MIB</i> . |
| dormant | |
| unknown | The port status is unknown. |
| disabled | The port was disabled. |

iplport_duplex

The operating duplex mode of the NetChange port, either *automatic* (*auto*) *half-duplex* (*half*) or *full-duplex* (*full*).

iplport_cfg_duplex

The configured duplex mode of the NetChange port, either *automatic* (*auto*) *half-duplex* (*half*) or *full-duplex* (*full*).

iplport_defaultmau

Internal use. Not documented.

iplport_maitypeplist

Internal use. Not documented.

iplport_modulenumber

Internal use. Not documented.

iplport_slotnumber

The slot number of the port.

iplport_portnumber

The number of the port.

iplport_dot1d

Internal use. Not documented.

iplport_port_security_max_mac

The maximum number of MAC addresses allowed to access the port. the port-security mode must be enabled.

iplport_port_security_action

The port-security action of the NetChange port. Possible values can be:

- For HP devices: *disable* (1), *sendTrap* (2) or *sendTrapAndDisablePort* (3).
- For Cisco devices: *shutdown* (4), *dropNotify* (5) or *drop* (6).
- For Juniper devices: *none* (1), *drop* (2), *alarm* (3) or *shutdown* (4).

iplport_interco

The interconnection status of the NetChange port. 1 indicates that interconnection is enabled.

The interconnection status can be forced if the parameter **iplport_staticinterco** is set to 1.

iplport_dev_count

The number of discovered items associated with the NetChange port.

iplport_staticinterco

The forced interconnection status of the NetChange port. 1 indicates that the interconnection status returned in the parameter **iplport_interco** is forced.

iplport_analysis

Internal use. Not documented.

iplport_display

Internal use. Not documented.

iplport_secure

Internal use. Not documented.

iplport_neighbour

Internal use. Not documented.

iplport_class_name

The name of the class applied to the port, it can be preceded by the class directory.

iplport_status_time

The last time the port status has changed to *Active*, in decimal UNIX date format.

iplport_is_aggregated

The aggregation status of the port. 1 indicates that the port is aggregated.

aggregated_iplport_id

The database identifier (ID) of the aggregated port associated with the port.

iplport_poe

The port power over ethernet status. 1 indicates that the port provides POE.

iplport_poe_pwr_max

The maximum power over ethernet of the NetChange port, in watts.

iplport_poe_pwr_allocated

The power over ethernet allocated to the NetChange port, in watts.

in_bw_1h

The size of the incoming bandwidth of the NetChange port during the last hour, in bps.

out_bw_1h

The size of the outgoing bandwidth of the NetChange port during the last hour, in bps.

in_errors_1h

The size of the incoming traffic in error of the NetChange port during the last hour, in bps.

out_errors_1h

The size of the outgoing traffic in error of the NetChange port during the last hour, in bps.

aggregated_port_name

The name of the aggregated port associated with the port.

iplport_id

The database identifier (ID) of the port.

snmp_id

Internal use. Not documented.

iplnetdev_ip_addr

The IPv4 address of the NetChange network device the object belongs to.

iplnetdev_ip6_addr

The IPv6 address of the NetChange network device the object belongs to.

iplnetdev_site_id

The database identifier (ID) of the space associated with the NetChange network device the object belongs to.

iplnetdev_stack_id

The stack identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_vendor

The vendor name of the NetChange network device the object belongs to.

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplnetdev_type

The product name of the NetChange network device the object belongs to.

iplnetdev_serial

The serial number of the NetChange network device the object belongs to.

iplnetdev_syscontact

The contact associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_nbports

The total number of ports the NetChange network device the object belongs to contains.

iplnetdev_status

The status of the NetChange network device the object belongs to, either *OK* (1), in *timeout* (2) or *misconfigured* (3).

iplnetdev_class_name

The name of the class applied to the NetChange network device the object belongs to, it can be preceded by the class directory.

iplnetdev_dot1x

The 802.1X authentication status of the NetChange network device the port belongs to, either *unsupported* (0), *enabled* (1) or *disabled* (2).

iplnetdev_syslocation

The physical location associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_port_security

The port-security status of the NetChange network device the port belongs to, either *unsupported* (0), *enabled* (1) or *disabled* (2).

iplnetdevvlan_name

The name of the NetChange VLAN the object belongs to.

iplnetdevvlan_number

The VLAN identifier (ID) of the NetChange VLAN the port belongs to.

iplport_class_parameters

The class parameters applied to the port and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

iplport_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **iplport_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....

iplport_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

iplnetdev_class_parameters

The class parameters applied to the NetChange network device the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

iplnetdev_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **iplnetdev_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

Name

iplport_info — Display the properties of a port

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

iplport_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

iplport_id

The database identifier (ID) of the port, a unique numeric key value automatically incremented when you add a port. Use the ID to specify the port of your choice.

Output Parameters

port_vlan_number_list

The list of the VLAN identifier (ID) of all the NetChange VLANs associated with the NetChange port, as follows: `<vlan_id>, <vlan_id>...`.

port_vlan_name_list

The list of the name of all the NetChange VLANs associated with the NetChange port, as follows: `<vlan_id>, <vlan_id>...`.

tagged_vlan_list

The list of the VLAN identifier (ID) of the tagged NetChange VLANs associated with the NetChange port, as follows: `<vlan_id>, <vlan_id>...`.

iplport_oper_dot1x

The operating 802.1X authentication status of the NetChange port, either *unsupported* (0), *enabled* (1) or *disabled* (2).

iplport_port_security

The port-security mode of the NetChange port, either *unsupported* (0), *disabled* (1), *FirstN* (2), *configureSpecific* (4), *8021xAUTHORIZED* (5) or *LimitedContinuous* (6).

iplport_dot1x

The configured 802.1X authentication status of the NetChange port, either *unsupported* (0), *enabled* (1) or *disabled* (2).

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplport_ifnumber

The interface number of the NetChange port, as follows: <slot_number>. <port_number>, or <port_number> only when the slot number is 0.

iplport_name

The name of the port.

iplport_description

The description of the port.

iplport_ifname

The internal monitoring name of the port.

iplport_iftype

The type of the NetChange port.

iplport_ifdescr

The internal monitoring description of the NetChange port.

iplport_ifaddress

Internal use. Not documented.

iplport_ifvlan

Internal use. Not documented.

iplport_untagged_vlan

The list of the VLAN identifier (ID) of the untagged NetChange VLANs associated with the NetChange port, as follows: <vlan_id>, <vlan_id>... .

iplport_vlan_tagging

The VLAN tagging status of the NetChange port, either *trunk* (1), *access* (2), *auto* (4), *mixed* (6) or *tagged* (7).

iplport_trunk_status

The VLAN trunking status of the NetChange port, either *N/A* (0), *Trunk/Tagged* (1), or *Access/Untagged* (2).

iplport_ifspeed_max

The maximum speed of the NetChange port, in bits per seconds (bps). 0 indicates that the port is not active.

iplport_ifvendor

The vendor name of the NetChange port.

iplport_ifoperspeed

The operating speed of the NetChange port, in bits per seconds (bps). 0 indicates that the port is not active.

iplport_ifcfgspeed

The configured speed of the NetChange port, in bits per seconds (bps). -1 indicates that the speed is automatically configured (*auto*).

iplport_ifoperstatus

The operational status of the port:

Table 57.2. iplport_ifoperstatus possible values

| Status | Description |
|----------------|---|
| active | The port is active, or up. |
| inactive | The port is inactive, or down. |
| testing | The port is up but no operational packets can be passed. |
| lowerLayerDown | |
| notPresent | The port is inactive. These statuses are very rare. For more details, refer to the description of the MIB <i>IF-MIB</i> . |
| dormant | |
| unknown | The port status is unknown. |
| disabled | The port was disabled. |

iplport_duplex

The operating duplex mode of the NetChange port, either *automatic* (*auto*) *half-duplex* (*half*) or *full-duplex* (*full*).

iplport_cfg_duplex

The configured duplex mode of the NetChange port, either *automatic* (*auto*) *half-duplex* (*half*) or *full-duplex* (*full*).

iplport_defaultmau

Internal use. Not documented.

iplport_mautypelist

Internal use. Not documented.

iplport_modulenumber

Internal use. Not documented.

iplport_slotnumber

The slot number of the port.

iplport_portnumber

The number of the port.

iplport_dot1d

Internal use. Not documented.

iplport_port_security_max_mac

The maximum number of MAC addresses allowed to access the port. the port-security mode must be enabled.

iplport_port_security_action

The port-security action of the NetChange port. Possible values can be:

- For HP devices: *disable* (1), *sendTrap* (2) or *sendTrapAndDisablePort* (3).
- For Cisco devices: *shutdown* (4), *dropNotify* (5) or *drop* (6).
- For Juniper devices: *none* (1), *drop* (2), *alarm* (3) or *shutdown* (4).

iplport_interco

The interconnection status of the NetChange port. 1 indicates that interconnection is enabled. The interconnection status can be forced if the parameter **iplport_staticinterco** is set to 1.

iplport_dev_count

The number of discovered items associated with the NetChange port.

iplport_staticinterco

The forced interconnection status of the NetChange port. 1 indicates that the interconnection status returned in the parameter **iplport_interco** is forced.

iplport_analysis

Internal use. Not documented.

iplport_display

Internal use. Not documented.

iplport_secure

Internal use. Not documented.

iplport_neighbour

Internal use. Not documented.

iplport_class_name

The name of the class applied to the port, it can be preceded by the class directory.

iplport_status_time

The last time the port status has changed to *Active*, in decimal UNIX date format.

iplport_is_aggregated

The aggregation status of the port. 1 indicates that the port is aggregated.

aggregated_iplport_id

The database identifier (ID) of the aggregated port associated with the port.

iplport_poe

The port power over ethernet status. 1 indicates that the port provides POE.

iplport_poe_pwr_max

The maximum power over ethernet of the NetChange port, in watts.

iplport_poe_pwr_allocated

The power over ethernet allocated to the NetChange port, in watts.

in_bw_1h

The size of the incoming bandwidth of the NetChange port during the last hour, in bps.

out_bw_1h

The size of the outgoing bandwidth of the NetChange port during the last hour, in bps.

in_errors_1h

The size of the incoming traffic in error of the NetChange port during the last hour, in bps.

out_errors_1h

The size of the outgoing traffic in error of the NetChange port during the last hour, in bps.

aggregated_port_name

The name of the aggregated port associated with the port.

iplport_id

The database identifier (ID) of the port.

snmp_id

Internal use. Not documented.

iplnetdev_ip_addr

The IPv4 address of the NetChange network device the object belongs to.

iplnetdev_ip6_addr

The IPv6 address of the NetChange network device the object belongs to.

iplnetdev_site_id

The database identifier (ID) of the space associated with the NetChange network device the object belongs to.

iplnetdev_stack_id

The stack identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_vendor

The vendor name of the NetChange network device the object belongs to.

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplnetdev_type

The product name of the NetChange network device the object belongs to.

iplnetdev_serial

The serial number of the NetChange network device the object belongs to.

iplnetdev_syscontact

The contact associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_nbports

The total number of ports the NetChange network device the object belongs to contains.

iplnetdev_status

The status of the NetChange network device the object belongs to, either *OK* (1), in *timeout* (2) or *misconfigured* (3).

iplnetdev_class_name

The name of the class applied to the NetChange network device the object belongs to, it can be preceded by the class directory.

iplnetdev_dot1x

The 802.1X authentication status of the NetChange network device the port belongs to, either *unsupported* (0), *enabled* (1) or *disabled* (2).

iplnetdev_syslocation

The physical location associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_port_security

The port-security status of the NetChange network device the port belongs to, either *unsupported* (0), *enabled* (1) or *disabled* (2).

iplnetdevvlan_name

The name of the NetChange VLAN the object belongs to.

iplnetdevvlan_number

The VLAN identifier (ID) of the NetChange VLAN the port belongs to.

iplport_class_parameters

The class parameters applied to the port and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

iplport_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **iplport_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

iplport_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

iplnetdev_class_parameters

The class parameters applied to the NetChange network device the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

iplnetdev_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **iplnetdev_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .

Name

iplport_count — Count the number of ports

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Chapter 58. NetChange IPv4 Address

Name

iplnetdevaddr_list — List the NetChange IPv4 Addresses

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter **offset** must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter **limit** must be specified in **lowercase**.

Output Parameters

iplnetdevaddr_id

The database identifier (ID) of the IPv4 address configured for the interface of the network device.

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplnetdevip_ip_addr

The IPv4 address of the interface configured on the device, in hexadecimal format.

iplnetdevip_ip_mask

The address of the network the IP address belongs to, in CIDR format.

iplnetdevip_ip_addrtype

The IP address protocol version, IPv4 (4) or IPv6 (6).

iplnetdevip_ip_status

The status of the IP address of the interface configured on the device, either active (1) or inactive (0).

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplport_id

The database identifier (ID) of the port the object belongs to.

iplnetdevip_name

The name of the IP address of the interface configured on the device.

iplnetdevip_vlan

The ID of the VLAN the IP address belongs to, as set on the network device.

iplnetdevip_vrfname

The name of the VRF, as set on the network device.

iplnetdevip_vrfrd

The Route Distinguisher of **iplnetdevip_vrfname**. A unique 64-bits identifier, that can be composed of IP addresses or AS Numbers, that differentiates every set of routes on each VRF.

Name

iplnetdevaddr_info — Display the properties of a NetChange IPv4 Address

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

iplnetdevaddr_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

iplnetdevaddr_id

The database identifier (ID) of the IPv4 address configured for the interface of the network device, a unique numeric key value automatically incremented when the interface was discovered on the device. Use it to identify the IP address of your choice.

Output Parameters

iplnetdevaddr_id

The database identifier (ID) of the IPv4 address configured for the interface of the network device.

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplnetdevip_ip_addr

The IPv4 address of the interface configured on the device, in hexadecimal format.

iplnetdevip_ip_mask

The address of the network the IP address belongs to, in CIDR format.

iplnetdevip_ip_addrtype

The IP address protocol version, IPv4 (4) or IPv6 (6).

iplnetdevip_ip_status

The status of the IP address of the interface configured on the device, either active (1) or inactive (0).

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplport_id

The database identifier (ID) of the port the object belongs to.

iplnetdevip_name

The name of the IP address of the interface configured on the device.

iplnetdevip_vlan

The ID of the VLAN the IP address belongs to, as set on the network device.

iplnetdevip_vrfname

The name of the VRF, as set on the network device.

iplnetdevip_vrfrd

The Route Distinguisher of **iplnetdevip_vrfname**. A unique 64-bits identifier, that can be composed of IP addresses or AS Numbers, that differentiates every set of routes on each VRF.

Name

iplnetdevaddr_count — Count the number of NetChange IPv4 Addresses

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

iplnetdevaddr_groupby — Group NetChange IPv4 Addresses by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

iplnetdevaddr_groupby_count — Count the number of NetChange IPv4 Addresses grouped by parameter(s)

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Chapter 59. NetChange IPv6 Address

Name

iplnetdevaddr6_list — List the NetChange IPv6 Addresses

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter **offset** must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter **limit** must be specified in **lowercase**.

Output Parameters

iplnetdevaddr_id

The database identifier (ID) of the IPv6 address configured for the interface of the network device.

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplnetdevip_ip_addr

The IPv6 address of the interface configured on the device, in hexadecimal format.

iplnetdevip_ip_mask

The address of the network the IP address belongs to, in CIDR format.

iplnetdevip_ip_addrtype

The IP address protocol version, IPv4 (4) or IPv6 (6).

iplnetdevip_ip_status

The status of the IP address of the interface configured on the device, either active (1) or inactive (0).

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplport_id

The database identifier (ID) of the port the object belongs to.

iplnetdevip_name

The name of the IP address of the interface configured on the device.

iplnetdevip_vlan

The ID of the VLAN the IP address belongs to, as set on the network device.

iplnetdevip_vrfname

The name of the VRF, as set on the network device.

iplnetdevip_vrfrd

The Route Distinguisher of **iplnetdevip_vrfname**. A unique 64-bits identifier, that can be composed of IP addresses or AS Numbers, that differentiates every set of routes on each VRF.

Name

iplnetdevaddr6_info — Display the properties of a NetChange IPv6 Address

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

iplnetdevaddr_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

iplnetdevaddr_id

The database identifier (ID) of the IPv6 address configured for the interface of the network device, a unique numeric key value automatically incremented when the interface was discovered on the device. Use it to identify the IP address of your choice.

Output Parameters

iplnetdevaddr_id

The database identifier (ID) of the IPv6 address configured for the interface of the network device.

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplnetdevip_ip_addr

The IPv6 address of the interface configured on the device, in hexadecimal format.

iplnetdevip_ip_mask

The address of the network the IP address belongs to, in CIDR format.

iplnetdevip_ip_addrtype

The IP address protocol version, IPv4 (4) or IPv6 (6).

iplnetdevip_ip_status

The status of the IP address of the interface configured on the device, either active (1) or inactive (0).

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplport_id

The database identifier (ID) of the port the object belongs to.

iplnetdevip_name

The name of the IP address of the interface configured on the device.

iplnetdevip_vlan

The ID of the VLAN the IP address belongs to, as set on the network device.

iplnetdevip_vrfname

The name of the VRF, as set on the network device.

iplnetdevip_vrfrd

The Route Distinguisher of **iplnetdevip_vrfname**. A unique 64-bits identifier, that can be composed of IP addresses or AS Numbers, that differentiates every set of routes on each VRF.

Name

iplnetdevaddr6_count — Count the number of NetChange IPv6 Addresses

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

iplnetdevaddr6_groupby — Group the NetChange IPv6 Addresses by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

iplnetdevaddr6_groupby_count — Count the number of NetChange IPv6 Addresses grouped by parameter(s)

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Chapter 60. Discovered Item

Name

ipldev_list — List the discovered items

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

ipldev_id

The database identifier (ID) of the discovered item.

iplport_id

The database identifier (ID) of the port the object belongs to.

ipldev_mac

The MAC address associated with the discovered item.

ipldev_vlan

The VLAN identifier (ID) of the NetChange VLAN associated with the discovered item.

ipldev_time

The time at which the item has been discovered, in decimal UNIX date format.

ipldev_end_time

The last time the item has been seen, in decimal UNIX date format.

ipldev_first_seen

The time and date, in decimal format, when the MAC address of the discovered item was seen for the first time on a port, VLAN or network device.

iplport_dev_count

The number of discovered items associated with the same NetChange port.

iplport_name

The name of the port the object belongs to.

iplport_ifnumber

The interface number of the NetChange port associated with the discovered item, as follows:
<slot_number>.<port_number>, or *<port_number>* only when the slot number is 0.

iplport_description

The description of the port the object belongs to.

iplport_slotnumber

The slot number of the port the discovered item is connected to.

iplport_portnumber

The number of the port the discovered item is connected to.

iplport_secure

Internal use. Not documented.

iplport_poe

The port power over ethernet status of the NetChange port associated with the discovered item. 1 indicates that the port associated with the discovered item provides POE.

iplport_class_name

The name of the class applied to the port the object belongs to, it can be preceded by the class directory.

iplport_interco

The interconnection status of the NetChange port associated with the discovered item. 1 indicates that interconnection is enabled. The interconnection status can be forced if the parameter **iplport_staticinterco** is set to 1.

iplport_staticinterco

The forced interconnection status of the NetChange port associated with the discovered item. 1 indicates that the interconnection status returned in the parameter **iplport_interco** is forced.

iplport_duplex

The operating duplex mode of the NetChange port associated with the discovered item, either *automatic (auto)* *half-duplex (half)* or *full-duplex (full)*.

iplport_iftype

The type of the NetChange port.

iplport_ifoperstatus

The operational status of the port the discovered item is connected to:

Table 60.1. *iplport_ifoperstatus* possible values

| Status | Description |
|----------------|---|
| active | The port is active, or up. |
| inactive | The port is inactive, or down. |
| testing | The port is up but no operational packets can be passed. |
| lowerLayerDown | |
| notPresent | The port is inactive. These statuses are very rare. For more details, refer to the description of the MIB <i>IF-MIB</i> . |
| dormant | |
| unknown | The port status is unknown. |
| disabled | The port was disabled. |

iplnetdev_id

The database identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_ip_addr

The IPv4 address of the NetChange network device the object belongs to.

iplnetdev_ip6_addr

The IPv6 address of the NetChange network device the object belongs to.

iplnetdev_hostaddr

The human readable version of the parameter **iplnetdev_ip_addr**.

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplnetdev_serial

The serial number of the NetChange network device the object belongs to.

iplnetdev_vendor

The vendor name of the NetChange network device the object belongs to.

iplnetdev_stack_id

The stack identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_type

The product name of the NetChange network device the object belongs to.

iplnetdev_nbports

The total number of ports the NetChange network device the object belongs to contains.

iplnetdev_syslocation

The physical location associated with the NetChange network device the object belongs to for SNMP monitoring.

iplnetdev_syscontact

The contact associated with the NetChange network device the object belongs to for SNMP monitoring.

mac_vendor

The vendor details of the discovered item.

iplnetdevvlan_name

The name of the NetChange VLAN the object belongs to.

iplnetdevvlan_status

The status of the NetChange VLAN associated with the discovered item, either *OK* (1), *inactive* (2) or *dynamic* (3).

hostiface_id

The database identifier (ID) of the Device Manager interface associated with the discovered item.

hostiface_name

The name of the Device Manager interface associated with the discovered item.

hostdev_id

The database identifier (ID) of the Device Manager device associated with the discovered item.

hostdev_name

The name of the Device Manager device associated with the discovered item.

iplvrf_id

The database identifier (ID) of the VRF that the discovered item belongs to.

iplvrf_name

The name of the VRF, as set on the network device the object belongs to.

iplvrf_rd

The Route Distinguisher of **iplvrf_name**. A unique 64-bits identifier, that can be composed of IP addresses or AS Numbers, that differentiates every set of routes on each VRF.

iplip_id

Internal use. Not documented.

iplip_ip_addr

The IPv4 address of the discovered item, in hexadecimal format.

iplip_ip6_addr

The IPv6 address of the discovered item, in hexadecimal format.

iplip_hostaddr

The human readable version of the parameter **iplip_ip_addr**.

iplip6_hostaddr

The human readable version of the parameter **iplip6_ip6_addr**.

iplip_source

The source of the discovered item, either *ARP* (1), *DHCP lease* (2), *DHCP static* (3) or *IPAM* (4), in this discovery order of priority.

iplip_dns_name

The name of the discovered item as automatically retrieved by NetChange if it is declared in an A or PTR record in one of your DNS servers.

Name

ipldev_count — Count the number of discovered items

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

ipldev_groupby — Group discovered items by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: `GROUPBY=<param1>,<param2>,...`. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement **SELECT** is returned.

Name

ipldev_groupby_count — Count the number of discovered items grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

ipldev_log_count — Count the number of discovered items moved to the *Discovered items history*

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

ipldev_log_list — List the discovered items moved to the *Discovered items history*

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

ipldev_id

The database identifier (ID) of the discovered item.

ipldev_time

The time at which the item has been discovered, in decimal UNIX date format.

ipldev_end_time

The last time the item has been seen, in decimal UNIX date format.

histo_state

Internal use. Not documented.

iplport_ifnumber

The interface number of the NetChange port associated with the discovered item, as follows:

<slot_number>.<port_number>, or <port_number> only when the slot number is 0.

iplport_name

The name of the port the object belongs to.

iplport_portnumber

The number of the port the discovered item is connected to.

iplport_slotnumber

The slot number of the port the discovered item is connected to.

ipldev_mac

The MAC address associated with the discovered item.

ipldev_vlan

The VLAN identifier (ID) of the NetChange VLAN associated with the discovered item.

delete_time

Internal use. Not documented.

iplip_ip_addr

The IPv4 address of the discovered item, in hexadecimal format.

iplip_ip6_addr

The IPv6 address of the discovered item, in hexadecimal format.

iplip_source

The source of the discovered item, either *ARP* (1), *DHCP lease* (2), *DHCP static* (3) or *IPAM* (4), in this discovery order of priority.

iplip_dns_name

The name of the discovered item as automatically retrieved by NetChange if it is declared in an A or PTR record in one of your DNS servers.

iplnetdev_name

The name of the NetChange network device the object belongs to.

iplnetdev_ip_addr

The IPv4 address of the NetChange network device the object belongs to.

iplnetdev_ip6_addr

The IPv6 address of the NetChange network device the object belongs to.

iplnetdev_stack_id

The stack identifier (ID) of the NetChange network device the object belongs to.

iplnetdev_type

The product name of the NetChange network device the object belongs to.

iplnetdev_site_id

The database identifier (ID) of the space associated with the NetChange network device the object belongs to.

mac_vendor

The vendor details of the discovered item.

iplnetdevvlan_name

The name of the NetChange VLAN the object belongs to.

iplvrf_name

The name of the VRF, as set on the network device the object belongs to.

iplvrf_rd

The Route Distinguisher of **iplvrf_name**. A unique 64-bits identifier, that can be composed of IP addresses or AS Numbers, that differentiates every set of routes on each VRF.

iplport_interco

The interconnection status of the NetChange port associated with the discovered item. 1 indicates that interconnection is enabled. The interconnection status can be forced if the parameter **iplport_staticinterco** is set to 1.

iplport_staticinterco

The forced interconnection status of the NetChange port associated with the discovered item. 1 indicates that the interconnection status returned in the parameter **iplport_interco** is forced.

Part X. Workflow Services

Table of Contents

| | |
|--------------------------------------|------|
| 61. Request | 1061 |
| workflow_request_add | 1062 |
| request_incoming_list | 1067 |
| request_incoming_info | 1071 |
| request_incoming_count | 1074 |
| request_incoming_groupby | 1075 |
| request_incoming_groupby_count | 1077 |
| request_outgoing_list | 1079 |
| request_outgoing_info | 1083 |
| request_outgoing_count | 1086 |
| request_outgoing_groupby | 1087 |
| request_outgoing_groupby_count | 1089 |

Chapter 61. Request

Name

workflow_request_add — Add a request

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

request_id

The database identifier (ID) of the Workflow request, a unique numeric key value automatically incremented when you add a request. Use the ID to specify the request of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

request_name

The name of the request, each request must have a unique name.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

request_object_type

The object of the request: *dnszone* for a DNS zone, *block* for an IPv4 block-type network, *subnet* for an IPv4 terminal subnet-type network, *pool* for an IPv4 pool or *ip* for an IPv4 address.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

request_object_id

The database identifier (ID) of the object of the request.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

request_object_name

The name of the object of the request.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

request_target_type

The type of group that deals with the request. The accepted values are: *group*, *DNS server* and *VLSM space*. If you do not set any value, the parameter is automatically set to *group*.

- Set it to *group* to assign it to a specific group of users, identified with the parameter **request_target_id**.
- Set it to *DNS server* to assign it to any group of users that has among its resources the DNS server specified with the parameter **request_target_name** or the parameter **request_target_id**.
- Set it to *VLSM space* to assign it to any group of users that has among its resources the IPAM space specified with the parameter **request_target_name** or the parameter **request_target_id**.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

request_target_name

The name of the resource associated with the parameter **request_target_type**.

- If you set the parameter **request_target_type** to *DNS server*, you can specify the name of a DNS server. Any group with the specified DNS server among its resource can execute the request.
- If you set the parameter **request_target_type** to *VLSM space*, you can specify the name of an IPAM space. Any group with the specified IPAM space among its resource can execute the request.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

request_target_id

The database identifier (ID) of the resource associated with the parameter **request_target_type**.

- If you set the parameter **request_target_type** to *group*, specify the identifier (ID) of the group of users that will deal with your request.
- If you set the parameter **request_target_type** to *DNS server*, you can specify the database identifier (ID) of a DNS server. Any group with the specified DNS server among its resource can execute the request.
- If you set the parameter **request_target_type** to *VLSM space*, you can specify the database identifier (ID) of an IPAM space. Any group with the specified IPAM space among its resource can execute the request.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

request_source_type

The type of resource associated with the parameter **request_source_name**. It is useless to set it when adding or editing a request, it is automatically set to *group*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

request_source_name

This optional parameter allows you to specify the name of a group of users when creating a Workflow request. It is useful if the requestor belongs to several groups. The rights of the specified group are used when executing the service. Besides, all the users of the specified

group can edit the request. The group specified can be displayed in the GUI in the column Source name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

request_source_id

This optional parameter allows you to specify the database identifier (ID) of a group of users when creating a Workflow request. It is useful if the requestor belongs to several groups. The rights of the specified group are used when executing the service. This parameter serves the same purpose as the parameter **request_source_name**.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

request_admin_id

The database identifier (ID) of the request manager, the user that deals with the request. It is useless to set it when adding or editing a request, it is automatically set when the request manager deals with the request.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

request_admin_time

The last time the request was edited by a request manager. It is useless to set it when adding or editing a request, it is automatically edited.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

modify_time

The last time the request was edited, by the requestor, a request manager or any other user with sufficient rights. It is useless to set it when adding or editing a request, it is automatically edited.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

request_status

The request status. By default, six statuses exist: *accept*, *archive*, *cancel*, *finish*, *handle*, *new* and *reject*. It is useless to set it when adding a request, it is automatically set to new. This parameter can also be returned and set with statuses that you or your administrator created.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | new | Can be edited | Yes |

request_action

The action required in the request. The accepted values are: *New*, *Modify* and *Delete*. They allow you to ask for the addition, editing or deletion of the object specified in the parameter **request_object_type**.

| | | | |
|------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
|------|--------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

request_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

request_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|----------------------|--------|-----------------------|------|
| Type | String | Maximum length | 4000 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

request_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... . If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.

- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

request_incoming_list — List the incoming requests

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter **offset** must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter **limit** must be specified in **lowercase**.

Output Parameters

request_id

The database identifier (ID) of the Workflow request.

oid

Internal use. Not documented.

request_name

The name of the request.

request_object_type

The object of the request: *dnszone* for a DNS zone, *block* for an IPv4 block-type network, *subnet* for an IPv4 terminal subnet-type network, *pool* for an IPv4 pool or *ip* for an IPv4 address.

request_object_id

The database identifier (ID) of the object of the request.

request_object_name

The name of the object of the request.

request_target_type

The type of group that deals with the request. The accepted values are: *group*, *DNS server* and *VLSM space*. If you did not set any value, the parameter is automatically set to *group*.

- When set to *group*, the request is assigned to a specific group of users, identified with the parameter **request_target_id**.
- When set to *DNS server*, the request is assigned to any group of users that has among its resources the DNS server specified with the parameter **request_target_name** or the parameter **request_target_id**.
- When set to *VLSM space*, the request is assigned to any group of users that has among its resources the IPAM space specified with the parameter **request_target_name** or the parameter **request_target_id**.

request_target_name

The name of the resource associated with the parameter **request_target_type**, either a *group*, a *DNS server* or a *VLSM space*.

request_target_id

The database identifier (ID) of the resource associated with the parameter **request_target_type**, either a *group*, a *DNS server* or a *VLSM space*.

request_source_type

The type of resource associated with the parameter **request_source_name**. It is always set to *group*.

request_source_name

The name of the group the user that created the request belongs to.

request_source_id

The database identifier (ID) of the group the user that created the request belongs to.

request_action

The action required in the request, either *New*, *Modify* or *Delete*.

request_usr_id

The database identifier (ID) of the last user that handled the request.

request_usr_time

The last time a user handled the request, in decimal UNIX date format.

request_admin_id

The database identifier (ID) of the request manager, the user that deals with the request, it is automatically set when the request manager deals with the request.

request_admin_time

The last time the user that created the request, i.e. the administrator of the request, handled it, in decimal UNIX date format.

request_status

The request status. By default, six statuses exist: *accept*, *archive*, *cancel*, *finish*, *handle*, *new* and *reject*. This parameter can also be returned and set with statuses that you or your administrator created.

request_class_name

The name of the class applied to the request, it can be preceded by the class directory.

create_time

The time the request was added, in decimal UNIX date format.

modify_time

The last time the request was edited, by the requestor, a request manager or any other user with sufficient rights, in decimal UNIX date format..

delete_time

The time the request was deleted, in decimal UNIX date format.

modif_tag

Internal use. Not documented.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

request_usr_login

The name of the last user that handled the request.

request_admin_login

The name of the user that created the request, i.e. the administrator of the request.

request_class_parameters

The class parameters applied to the request and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

request_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **request_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

request_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:
<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

Name

request_incoming_info — Display the properties of an incoming request

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

request_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

request_id

The database identifier (ID) of the Workflow request, a unique numeric key value automatically incremented when you add a request. Use the ID to specify the request of your choice.

Output Parameters

request_id

The database identifier (ID) of the Workflow request.

oid

Internal use. Not documented.

request_name

The name of the request.

request_object_type

The object of the request: *dnszone* for a DNS zone, *block* for an IPv4 block-type network, *subnet* for an IPv4 terminal subnet-type network, *pool* for an IPv4 pool or *ip* for an IPv4 address.

request_object_id

The database identifier (ID) of the object of the request.

request_object_name

The name of the object of the request.

request_target_type

The type of group that deals with the request. The accepted values are: *group*, *DNS server* and *VLSM space*. If you did not set any value, the parameter is automatically set to *group*.

- When set to *group*, the request is assigned to a specific group of users, identified with the parameter **request_target_id**.
- When set to *DNS server*, the request is assigned to any group of users that has among its resources the DNS server specified with the parameter **request_target_name** or the parameter **request_target_id**.
- When set to *VLSM space*, the request is assigned to any group of users that has among its resources the IPAM space specified with the parameter **request_target_name** or the parameter **request_target_id**.

request_target_name

The name of the resource associated with the parameter **request_target_type**, either a *group*, a *DNS server* or a *VLSM space*.

request_target_id

The database identifier (ID) of the resource associated with the parameter **request_target_type**, either a *group*, a *DNS server* or a *VLSM space*.

request_source_type

The type of resource associated with the parameter **request_source_name**. It is always set to *group*.

request_source_name

The name of the group the user that created the request belongs to.

request_source_id

The database identifier (ID) of the group the user that created the request belongs to.

request_action

The action required in the request, either *New*, *Modify* or *Delete*.

request_usr_id

The database identifier (ID) of the last user that handled the request.

request_usr_time

The last time a user handled the request, in decimal UNIX date format.

request_admin_id

The database identifier (ID) of the request manager, the user that deals with the request, it is automatically set when the request manager deals with the request.

request_admin_time

The last time the user that created the request, i.e. the administrator of the request, handled it, in decimal UNIX date format.

request_status

The request status. By default, six statuses exist: *accept*, *archive*, *cancel*, *finish*, *handle*, *new* and *reject*. This parameter can also be returned and set with statuses that you or your administrator created.

request_class_name

The name of the class applied to the request, it can be preceded by the class directory.

create_time

The time the request was added, in decimal UNIX date format.

modify_time

The last time the request was edited, by the requestor, a request manager or any other user with sufficient rights, in decimal UNIX date format..

delete_time

The time the request was deleted, in decimal UNIX date format.

modif_tag

Internal use. Not documented.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

request_usr_login

The name of the last user that handled the request.

request_admin_login

The name of the user that created the request, i.e. the administrator of the request.

request_class_parameters

The class parameters applied to the request and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

request_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **request_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

request_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

request_incoming_count — Count the number of incoming requests

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

request_incoming_groupby — Group incoming requests by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: <parameter>='<value>'. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

request_incoming_groupby_count — Count the number of incoming requests grouped by parameter(s)

Description

This service allows you to display the total number of results of the service `*_groupby`.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement `GROUPBY`.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: `count`, `max`, `min`, `sum` or `avg`. The aggregation function syntax is the following: `SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)` where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement `SELECT` must also be specified in the statement `GROUPBY`.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement `SELECT` without aggregation function must be specified in the statement `GROUPBY`.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

request_outgoing_list — List the outgoing requests

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter **offset** must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter **limit** must be specified in **lowercase**.

Output Parameters

request_id

The database identifier (ID) of the Workflow request.

oid

Internal use. Not documented.

request_name

The name of the request.

request_object_type

The object of the request: *dnszone* for a DNS zone, *block* for an IPv4 block-type network, *subnet* for an IPv4 terminal subnet-type network, *pool* for an IPv4 pool or *ip* for an IPv4 address.

request_object_id

The database identifier (ID) of the object of the request.

request_object_name

The name of the object of the request.

request_target_type

The type of group that deals with the request. The accepted values are: *group*, *DNS server* and *VLSM space*. If you did not set any value, the parameter is automatically set to *group*.

- When set to *group*, the request is assigned to a specific group of users, identified with the parameter **request_target_id**.
- When set to *DNS server*, the request is assigned to any group of users that has among its resources the DNS server specified with the parameter **request_target_name** or the parameter **request_target_id**.
- When set to *VLSM space*, the request is assigned to any group of users that has among its resources the IPAM space specified with the parameter **request_target_name** or the parameter **request_target_id**.

request_target_name

The name of the resource associated with the parameter **request_target_type**, either a *group*, a *DNS server* or a *VLSM space*.

request_target_id

The database identifier (ID) of the resource associated with the parameter **request_target_type**, either a *group*, a *DNS server* or a *VLSM space*.

request_source_type

The type of resource associated with the parameter **request_source_name**. It is always set to *group*.

request_source_name

The name of the group the user that created the request belongs to.

request_source_id

The database identifier (ID) of the group the user that created the request belongs to.

request_action

The action required in the request, either *New*, *Modify* or *Delete*.

request_usr_id

The database identifier (ID) of the last user that handled the request.

request_usr_time

The last time a user handled the request, in decimal UNIX date format.

request_admin_id

The database identifier (ID) of the request manager, the user that deals with the request, it is automatically set when the request manager deals with the request.

request_admin_time

The last time the user that created the request, i.e. the administrator of the request, handled it, in decimal UNIX date format.

request_status

The request status. By default, six statuses exist: *accept*, *archive*, *cancel*, *finish*, *handle*, *new* and *reject*. This parameter can also be returned and set with statuses that you or your administrator created.

request_class_name

The name of the class applied to the request, it can be preceded by the class directory.

create_time

The time the request was added, in decimal UNIX date format.

modify_time

The last time the request was edited, by the requestor, a request manager or any other user with sufficient rights, in decimal UNIX date format..

delete_time

The time the request was deleted, in decimal UNIX date format.

modif_tag

Internal use. Not documented.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

request_usr_login

The name of the last user that handled the request.

request_admin_login

The name of the user that created the request, i.e. the administrator of the request.

request_class_parameters

The class parameters applied to the request and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

request_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **request_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

request_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:
<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

Name

request_outgoing_info — Display the properties of an outgoing request

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

request_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

request_id

The database identifier (ID) of the Workflow request, a unique numeric key value automatically incremented when you add a request. Use the ID to specify the request of your choice.

Output Parameters

request_id

The database identifier (ID) of the Workflow request.

oid

Internal use. Not documented.

request_name

The name of the request.

request_object_type

The object of the request: *dnszone* for a DNS zone, *block* for an IPv4 block-type network, *subnet* for an IPv4 terminal subnet-type network, *pool* for an IPv4 pool or *ip* for an IPv4 address.

request_object_id

The database identifier (ID) of the object of the request.

request_object_name

The name of the object of the request.

request_target_type

The type of group that deals with the request. The accepted values are: *group*, *DNS server* and *VLSM space*. If you did not set any value, the parameter is automatically set to *group*.

- When set to *group*, the request is assigned to a specific group of users, identified with the parameter **request_target_id**.
- When set to *DNS server*, the request is assigned to any group of users that has among its resources the DNS server specified with the parameter **request_target_name** or the parameter **request_target_id**.
- When set to *VLSM space*, the request is assigned to any group of users that has among its resources the IPAM space specified with the parameter **request_target_name** or the parameter **request_target_id**.

request_target_name

The name of the resource associated with the parameter **request_target_type**, either a *group*, a *DNS server* or a *VLSM space*.

request_target_id

The database identifier (ID) of the resource associated with the parameter **request_target_type**, either a *group*, a *DNS server* or a *VLSM space*.

request_source_type

The type of resource associated with the parameter **request_source_name**. It is always set to *group*.

request_source_name

The name of the group the user that created the request belongs to.

request_source_id

The database identifier (ID) of the group the user that created the request belongs to.

request_action

The action required in the request, either *New*, *Modify* or *Delete*.

request_usr_id

The database identifier (ID) of the last user that handled the request.

request_usr_time

The last time a user handled the request, in decimal UNIX date format.

request_admin_id

The database identifier (ID) of the request manager, the user that deals with the request, it is automatically set when the request manager deals with the request.

request_admin_time

The last time the user that created the request, i.e. the administrator of the request, handled it, in decimal UNIX date format.

request_status

The request status. By default, six statuses exist: *accept*, *archive*, *cancel*, *finish*, *handle*, *new* and *reject*. This parameter can also be returned and set with statuses that you or your administrator created.

request_class_name

The name of the class applied to the request, it can be preceded by the class directory.

create_time

The time the request was added, in decimal UNIX date format.

modify_time

The last time the request was edited, by the requestor, a request manager or any other user with sufficient rights, in decimal UNIX date format..

delete_time

The time the request was deleted, in decimal UNIX date format.

modif_tag

Internal use. Not documented.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

request_usr_login

The name of the last user that handled the request.

request_admin_login

The name of the user that created the request, i.e. the administrator of the request.

request_class_parameters

The class parameters applied to the request and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

request_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **request_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

request_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

request_outgoing_count — Count the number of outgoing requests

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

request_outgoing_groupby — Group outgoing requests by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

request_outgoing_groupby_count — Count the number of outgoing requests grouped by parameter(s)

Description

This service allows you to display the total number of results of the service `*_groupby`.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement `GROUPBY`.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: `count`, `max`, `min`, `sum` or `avg`. The aggregation function syntax is the following: `SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)` where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement `SELECT` must also be specified in the statement `GROUPBY`.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement `SELECT` without aggregation function must be specified in the statement `GROUPBY`.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Part XI. Device Manager Services

Table of Contents

| | |
|---------------------------------|------|
| 62. Device Manager Device | 1093 |
| hostdev_add | 1094 |
| hostdev_list | 1098 |
| hostdev_info | 1101 |
| hostdev_count | 1103 |
| hostdev_groupby | 1104 |
| hostdev_groupby_count | 1106 |
| hostdev_delete | 1108 |
| 63. Port and Interface | 1110 |
| hostiface_add | 1111 |
| hostiface_list | 1115 |
| hostiface_info | 1119 |
| hostiface_count | 1122 |
| hostiface_groupby | 1123 |
| hostiface_groupby_count | 1125 |
| link_hostiface_add | 1127 |
| link_hostiface_list | 1130 |
| link_hostiface_count | 1132 |
| link_hostiface_delete | 1133 |
| hostiface_delete | 1135 |

Chapter 62. Device Manager Device

Name

hostdev_add — Add a Device Manager device

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (hostdev_name)
- **Editing:** (hostdev_id || hostdev_name)

Input Parameters

hostdev_id

The database identifier (ID) of the Device Manager device, a unique numeric key value automatically incremented when you add a device. Use the ID to specify the device of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostdev_name

The name of the Device Manager device, each device must have a unique name.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostdev_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | | Can be edited | Yes |

hostdev_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: *<class-parameter1>=<value1>&<class-parameter2>=<value2>&... .*

| Type | String | Maximum length | 4000 |
|---------------|--------|----------------|------|
| Default value | | Can be edited | Yes |

hostdev_site_id

The database identifier (ID) of a space you want to associate with the Device Manager device.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostdev_ip_addr

The IP address you want to associate with the Device Manager device, in hexadecimal format.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostdev_addr

The IP address you want to associate with the Device Manager device, in decimal format.

| | | | |
|---------------|--------------|----------------|-----|
| Type | IPv4 address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ipInetdev_id

The database identifier (ID) of a NetChange network device you want to associate with the Device Manager device.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

row_enabled

The object activation status.

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.
- If set to 2, the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

| | | | |
|---------------|--------------------------|----------------|-----|
| Type | Fixed value: 1 2 3 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

hostdev_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: **<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....**. If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

hostdev_list — List the Device Manager devices

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter **offset** must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter **limit** must be specified in **lowercase**.

Output Parameters

hostdev_id

The database identifier (ID) of the Device Manager device.

hostdev_name

The name of the Device Manager device.

hostdev_class_name

The name of the class applied to the device, it can be preceded by the class directory.

iplnetdev_id

The database identifier (ID) of the NetChange network device associated with the Device Manager device.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, **row_enabled** is set to 1 when an object is created.

iplnetdev_name

The name of the NetChange network device associated with the Device Manager device.

hostdev_ip_addr

The IP address associated with the Device Manager device.

hostdev_ip_formated

The human readable version of the parameter **hostdev_ip_addr**.

hostdev_site_id

The database identifier (ID) of the space associated with the Device Manager device.

hostdev_site_name

The name of the space associated with the Device Manager device.

port_total

The total number of ports on the Device Manager device.

port_used

The number of ports on the Device Manager device that are currently active.

port_used_percent

The percentage of ports on the Device Manager device that are currently active.

port_free

The number of ports on the Device Manager device that are currently free.

iface_total

The total number of interfaces on the Device Manager device.

iface_used

The number of interfaces on the Device Manager device that are currently active.

iface_used_percent

The percentage of interfaces on the Device Manager device that are currently active.

iface_free

The number of interfaces on the Device Manager device that are currently free.

hostdev_class_parameters

The class parameters applied to the device and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

hostdev_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **hostdev_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

hostdev_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

hostdev_info — Display the properties of a Device Manager device

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

hostdev_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

hostdev_id

The database identifier (ID) of the Device Manager device, a unique numeric key value automatically incremented when you add a device. Use the ID to specify the device of your choice.

Output Parameters

hostdev_id

The database identifier (ID) of the Device Manager device.

hostdev_name

The name of the Device Manager device.

hostdev_class_name

The name of the class applied to the device, it can be preceded by the class directory.

iplnetdev_id

The database identifier (ID) of the NetChange network device associated with the Device Manager device.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.

- 1 indicates the object is enabled and managed.
 - 2 indicates the object is unmanaged, disabled or both depending on the context.
- By default, *row_enabled* is set to 1 when an object is created.

iplnetdev_name

The name of the NetChange network device associated with the Device Manager device.

hostdev_ip_addr

The IP address associated with the Device Manager device.

hostdev_ip_formated

The human readable version of the parameter **hostdev_ip_addr**.

hostdev_site_id

The database identifier (ID) of the space associated with the Device Manager device.

hostdev_site_name

The name of the space associated with the Device Manager device.

port_total

The total number of ports on the Device Manager device.

port_used

The number of ports on the Device Manager device that are currently active.

port_used_percent

The percentage of ports on the Device Manager device that are currently active.

port_free

The number of ports on the Device Manager device that are currently free.

iface_total

The total number of interfaces on the Device Manager device.

iface_used

The number of interfaces on the Device Manager device that are currently active.

iface_used_percent

The percentage of interfaces on the Device Manager device that are currently active.

iface_free

The number of interfaces on the Device Manager device that are currently free.

hostdev_class_parameters

The class parameters applied to the device and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

hostdev_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **hostdev_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

hostdev_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

hostdev_count — Count the number of Device Manager devices

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

hostdev_groupby — Group Device Manager devices by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: `GROUPBY=<param1>,<param2>,...`. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement `SELECT` is returned.

Name

hostdev_groupby_count — Count the number of Device Manager devices grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

hostdev_delete — Delete a Device Manager device

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(hostdev_id || hostdev_name)

Input Parameters

hostdev_id

The database identifier (ID) of the Device Manager device, a unique numeric key value automatically incremented when you add a device. Use the ID to specify the device of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostdev_name

The name of the Device Manager device.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 63. Port and Interface

Name

hostiface_add — Add a Device Manager port or interface

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (hostiface_name && (hostdev_id || hostdev_name) && hostiface_type
- **Editing:** (hostiface_id || (hostiface_name && (hostdev_id || hostdev_name))

Input Parameters

hostiface_id

The database identifier (ID) of the Device Manager port or interface, a unique numeric key value automatically incremented when you add a port or interface. Use the ID to specify the port or interface of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostdev_id

The database identifier (ID) of the Device Manager device, a unique numeric key value automatically incremented when you add a device. Use the ID to specify the device of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostdev_name

The name of the Device Manager device.

| | | | |
|---------------|-----|----------------|-----|
| Type | s | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface_name

The name of the Device Manager port or interface, each port or interface must have a unique name.

| | | | |
|---------------|-----|----------------|-----|
| Type | s | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface_type

A way to indicate if the object is either a *port* or an *interface*.

| | | | |
|---------------|--------------------------------|----------------|-----|
| Type | Fixed value: interface port | Maximum length | N/A |
| Default value | N/A | Can be edited | No |

hostiface_mac

The MAC address you want to associate with the Device Manager port or interface.

| | | | |
|---------------|-------------|----------------|-----|
| Type | MAC address | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface_addr

The IP addresses you want to associate with the Device Manager port or interface, as follows <ip4_list>,<ip6_list>.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

site_id

The database identifier (ID) of a space you want to associate with the Device Manager port or interface.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

ip4_list

The list of IPv4 addresses you want to associate to the Device Manager port or interface, in hexadecimal format, as follows: <ip_address>, <ip_address>,...

| | | | |
|---------------|--------------------------------|----------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

ip6_list

The list of IPv6 addresses you want to associate to the Device Manager port or interface, in hexadecimal format, as follows: <ip_address>, <ip_address>,...

| | | | |
|---------------|--------------------------------|----------------|-----|
| Type | List of strings separated by ; | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

iplport_id

The database identifier (ID) of a NetChange port you want to associate with the Device Manager port or interface.

| | | | |
|---------------|-----|----------------|-----|
| Type | =>0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

hostiface_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|-----|----------------|-----|
| Type | s | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|---|----------------|-----|
| Type | s | Maximum length | N/A |
| Default value | | Can be edited | Yes |

row_enabled

The object activation status.

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.
- If set to 2, the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... . Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... . If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.

- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

hostiface_list — List the Device Manager ports & interfaces

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

hostdev_id

The database identifier (ID) of the Device Manager device the object belongs to.

hostdev_name

The name of the Device Manager device.

hostdev_class_name

The name of the class applied to the device the object belongs to, it can be preceded by the class directory.

iplnetdev_id

The database identifier (ID) of the NetChange network device associated with the Device Manager device the object belongs to.

hostiface_id

The database identifier (ID) of the Device Manager port or interface.

hostiface_name

The name of the Device Manager port or interface.

hostiface_type

A way to indicate if the object is either a *port* or an *interface*.

hostiface_class_name

The name of the class applied to the port or interface, it can be preceded by the class directory.

pear_iface_id

The database identifier (ID) of the Device Manager port or interface linked with *hostiface_id*.

pear_ipl_iface_id

The database identifier (ID) of the NetChange port linked with *hostiface_id*.

iplport_id

The database identifier (ID) of the NetChange port associated with the Device Manager port or interface.

custom_db_data_id

The database identifier (ID) of the Custom Database entry associated with the Device Manager port or interface.

hostiface_mac

The MAC address associated with the Device Manager port or interface.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
 - 1 indicates the object is enabled and managed.
 - 2 indicates the object is unmanaged, disabled or both depending on the context.
- By default, *row_enabled* is set to 1 when an object is created.

add_time

The time at which the Device Manager port or interface has been added, in decimal UNIX date format.

modify_time

The last time the Device Manager port or interface data was reconciled, in decimal UNIX date format.

vendor_key

Internal use. Not documented.

vendor_mac

The vendor details of the client associated with the Device Manager port or interface.

iplnetdev_name

The name of the NetChange network device associated with the Device Manager device the object belongs to.

iplnetdev_enabled

The activation status of the NetChange network device associated with the DM port or interface.

- If set to 1, the NetChange network device is enabled and managed.
- If set to 2, the NetChange network device is unmanaged, disabled or both depending on the context.

iplport_name

The name of the NetChange port associated with the DM port or interface.

iplport_enabled

The activation status of the NetChange port associated with the DM port or interface.

- If set to 1, the NetChange port is enabled and managed.
- If set to 2, the NetChange port is unmanaged, disabled or both depending on the context.

hostiface_ip_addr

The IP address associated with the Device Manager port or interface.

hostiface_ip_formated

The human readable version of the parameter **hostiface_ip_addr**.

hostiface_site_id

The database identifier (ID) of the space associated with the Device Manager port or interface.

hostiface_site_name

The name of the space associated with the Device Manager port or interface.

hostiface_manual_link

The DM device and port or interface to which the object is manually linked, as follows:
<hostdev_name> (<hostiface_name>)

hostiface_auto_link

The DM device and port or interface to which the object is automatically linked, as follows:
<hostdev_name> (<hostiface_name>)

nb_ip

The number of IP addresses associated with the Device Manager port or interface.

hostiface_class_parameters

The class parameters applied to the port or interface and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

hostiface_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **hostiface_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

hostiface_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

hostdev_class_parameters

The class parameters applied to the Device Manager device the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

hostdev_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **hostdev_class_parameters**: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&.... .

Name

hostiface_info — Display the properties of a Device Manager port or interface

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

hostiface_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

hostiface_id

The database identifier (ID) of the Device Manager port or interface, a unique numeric key value automatically incremented when you add a port or interface. Use the ID to specify the port or interface of your choice.

Output Parameters

hostdev_id

The database identifier (ID) of the Device Manager device the object belongs to.

hostdev_name

The name of the Device Manager device.

hostdev_class_name

The name of the class applied to the device the object belongs to, it can be preceded by the class directory.

iplnetdev_id

The database identifier (ID) of the NetChange network device associated with the Device Manager device the object belongs to.

hostiface_id

The database identifier (ID) of the Device Manager port or interface.

hostiface_name

The name of the Device Manager port or interface.

hostiface_type

A way to indicate if the object is either a *port* or an *interface*.

hostiface_class_name

The name of the class applied to the port or interface, it can be preceded by the class directory.

pear_iface_id

The database identifier (ID) of the Device Manager port or interface linked with *hostiface_id*.

pear_ipl_iface_id

The database identifier (ID) of the NetChange port linked with *hostiface_id*.

iplport_id

The database identifier (ID) of the NetChange port associated with the Device Manager port or interface.

custom_db_data_id

The database identifier (ID) of the Custom Database entry associated with the Device Manager port or interface.

hostiface_mac

The MAC address associated with the Device Manager port or interface.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

add_time

The time at which the Device Manager port or interface has been added, in decimal UNIX date format.

modify_time

The last time the Device Manager port or interface data was reconciled, in decimal UNIX date format.

vendor_key

Internal use. Not documented.

vendor_mac

The vendor details of the client associated with the Device Manager port or interface.

iplnetdev_name

The name of the NetChange network device associated with the Device Manager device the object belongs to.

iplnetdev_enabled

The activation status of the NetChange network device associated with the DM port or interface.

- If set to 1, the NetChange network device is enabled and managed.
- If set to 2, the NetChange network device is unmanaged, disabled or both depending on the context.

iplport_name

The name of the NetChange port associated with the DM port or interface.

iplport_enabled

The activation status of the NetChange port associated with the DM port or interface.

- If set to 1, the NetChange port is enabled and managed.
- If set to 2, the NetChange port is unmanaged, disabled or both depending on the context.

hostiface_ip_addr

The IP address associated with the Device Manager port or interface.

hostiface_ip_formated

The human readable version of the parameter **hostiface_ip_addr**.

hostiface_site_id

The database identifier (ID) of the space associated with the Device Manager port or interface.

hostiface_site_name

The name of the space associated with the Device Manager port or interface.

hostiface_manual_link

The DM device and port or interface to which the object is manually linked, as follows:

<hostdev_name> (<hostiface_name>)

hostiface_auto_link

The DM device and port or interface to which the object is automatically linked, as follows:

<hostdev_name> (<hostiface_name>)

nb_ip

The number of IP addresses associated with the Device Manager port or interface.

hostiface_class_parameters

The class parameters applied to the port or interface and their value: *<class-parameter1>=<value1>&<class-parameter2>=<value2>&....*

hostiface_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **hostiface_class_parameters**: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....*

hostiface_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: *<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&....*

hostdev_class_parameters

The class parameters applied to the Device Manager device the object belongs to and their value: *<class-parameter1>=<value1>&<class-parameter2>=<value2>&....*

hostdev_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **hostdev_class_parameters**: *<classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&....*

Name

hostiface_count — Count the number Device Manager ports & interfaces

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

hostiface_groupby — Group Device Manager ports and interfaces by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: *GROUPBY=<param1>,<param2>,...*. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: *<parameter>='<value>'*. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement *SELECT* is returned.

Name

hostiface_groupby_count — Count the number of Device Manager ports and interfaces grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

link_hostiface_add — Link two Device Manager devices using their ports and/or interfaces

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** ((hostiface1_id || (hostiface1_name && hostdev1_name)) && (hostiface2_id || (hostiface2_name && hostdev2_name)))
- **Editing:** (link_hostiface_id || ((hostiface1_id || (hostiface1_name && hostdev1_name)) && (hostiface2_id || (hostiface2_name && hostdev2_name))))

Input Parameters

link_hostiface_id

The database identifier (ID) of the Device Manager port or interface link, a unique numeric key value automatically incremented when you add a link between a device and a port or interface. Use the ID to specify the port or interface link of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface1_id

The database identifier (ID) of the DM port or interface you want to link with *hostiface2_id*.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface2_id

The database identifier (ID) of the DM port or interface you want to link with *hostiface1_id*.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface1_name

The name of the DM port or interface you want to link with *hostiface2_id*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface2_name

The name of the DM port or interface you want to link with *hostiface1_id*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostdev1_name

The name of the device to which belongs the DM port or interface you want to link with *hostiface2_id*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostdev2_name

The name of the device to which belongs the DM port or interface you want to link with *hostiface1_id*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

auto_link

A way to determine if the link between two Device Manager devices is set automatically (1) or manually (0).

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | 0 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.

- *Notice:* the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

link_hostiface_list — List Device Manager ports & interfaces

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

hostiface_id

The database identifier (ID) of the Device Manager port or interface, a unique numeric key value automatically incremented when you add a port or interface. Use the ID to specify the port or interface of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

link_hostiface_id

The database identifier (ID) of the Device Manager port or interface link. Use the ID to specify the port or interface link of your choice.

hostiface_id

The database identifier (ID) of the Device Manager port or interface.

auto_link

A way to determine if the link between two Device Manager devices is set automatically (1) or manually (0).

hostiface_name

The name of the Device Manager port or interface.

hostiface_type

A way to indicate if the object is either a *port* or an *interface*.

hostiface_mac

The MAC address associated with the Device Manager port or interface.

add_time

The time at which the Device Manager port or interface has been added, in decimal UNIX date format.

hostdev_name

The name of the Device Manager device.

hostdev_id

The database identifier (ID) of the Device Manager device the object belongs to.

Name

link_hostiface_count — Count the number of links between Device Manager devices

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

hostiface_id

The database identifier (ID) of the Device Manager port or interface, a unique numeric key value automatically incremented when you add a port or interface. Use the ID to specify the port or interface of your choice.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

link_hostiface_delete — Delete a link between two Device Manager devices

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(link_hostiface_id || ((hostiface1_id || (hostiface1_name && hostdev1_name)) && (hostiface2_id || (hostiface2_name && hostdev2_name))))

Input Parameters

link_hostiface_id

The database identifier (ID) of the Device Manager port or interface link, a unique numeric key value automatically incremented when you add a link between a device and a port or interface. Use the ID to specify the port or interface link of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface1_id

The database identifier (ID) of the DM port or interface you want to unlink from *hostiface2_id*.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface2_id

The database identifier (ID) of the DM port or interface you want to unlink from *hostiface1_id*.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface1_name

The name of the DM port or interface you want to unlink from *hostiface2_id*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostiface2_name

The name of the DM port or interface you want to unlink from *hostiface1_id*.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostdev1_name

The name of the device to which belongs the DM port or interface you want to unlink from *hostiface2_id*.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

hostdev2_name

The name of the device to which belongs the DM port or interface you want to unlink from *hostiface1_id*.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

auto_link

A way to determine if the link between two Device Manager devices is set automatically (1) or manually (0).

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

hostiface_delete — Delete a Device Manager port or interface

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(hostiface_id || (hostiface_name && (hostdev_id || hostdev_name)))

Input Parameters

hostiface_id

The database identifier (ID) of the Device Manager port or interface, a unique numeric key value automatically incremented when you add a port or interface. Use the ID to specify the port or interface of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostdev_id

The database identifier (ID) of the Device Manager device, a unique numeric key value automatically incremented when you add a device. Use the ID to specify the device of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostdev_name

The name of the Device Manager device.

| Type | s | Maximum length | N/A |
|---------------|-----|----------------|-----|
| Default value | N/A | Can be edited | Yes |

hostiface_name

The name of the Device Manager port or interface.

| Type | s | Maximum length | N/A |
|---------------|-----|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Part XII. VLAN Manager Services

Table of Contents

| | |
|------------------------------|------|
| 64. VLAN Domain | 1139 |
| vlm_domain_add | 1140 |
| vlmdomain_list | 1143 |
| vlmdomain_info | 1145 |
| vlmdomain_count | 1147 |
| group_vlmdomain_add | 1148 |
| group_vlmdomain_delete | 1150 |
| vlm_domain_delete | 1152 |
| 65. VLAN Range | 1154 |
| vlm_range_add | 1155 |
| vlmrangle_list | 1159 |
| vlmrangle_info | 1162 |
| vlmrangle_count | 1164 |
| group_vlmrange_add | 1165 |
| group_vlmrange_delete | 1167 |
| vlm_range_delete | 1169 |
| 66. VLAN | 1171 |
| vlm_vlan_add | 1172 |
| vlmvlan_list | 1176 |
| vlmvlan_info | 1179 |
| vlmvlan_count | 1182 |
| vlm_vlan_delete | 1183 |

Chapter 64. VLAN Domain

Name

vlm_domain_add — Add/Edit a VLAN domain

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** vlmdomain_name
- **Editing:** (vlmdomain_id || vlmdomain_name)

Input Parameters

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_name

The name of the VLAN domain, each VLAN domain must have a unique name.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_description

The description of the VLAN domain.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | | Can be edited | Yes |

vlmdomain_start_vlan_id

The VLAN identifier (ID) of a VLAN, a numeric value between 1 and 4094. Use the ID to specify the first VLAN in the VLAN domain.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | 1 | Can be edited | Yes |

vlmdomain_end_vlan_id

The VLAN identifier (ID) of a VLAN, a numeric value between 1 and 4094. Use the ID to specify the last VLAN in the VLAN domain.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

support_vxlan

The type of virtual network used by the domain. Set it to *1* to use VXLAN or *0* to use VLAN.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 0 | Can be edited | No |

vlmdomain_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

vlmdomain_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: *<class-parameter1>=<value1>&<class-parameter2>=<value2>&....*

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: *<class-parameter1>&<class-parameter2>&....*. Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmdomain_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....* If the inheritance or propagation property is not specified, its default value - set, *propagate* - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

vlmdomain_list — List the VLAN domains

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

vlmdomain_id

The database identifier (ID) of the VLAN domain.

vlmdomain_name

The name of the VLAN domain.

vlmdomain_start_vlan_id

The VLAN identifier (ID) of the first VLAN in the VLAN domain.

vlmdomain_end_vlan_id

The VLAN identifier (ID) of the last VLAN in the VLAN domain.

vlmdomain_description

The description of the VLAN domain.

vlmdomain_class_name

The name of the class applied to the VLAN domain, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

support_vxlan

The type of virtual network used by the domain, VXLAN (1) or VLAN (0).

vlmdomain_class_parameters

The class parameters applied to the VLAN domain and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

vlmdomain_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **vlmdomain_class_parameters**: <classparam1>=<inheritance>, <propagation>&<classparam2>=<inheritance>, <propagation>&....

vlmdomain_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&....

Name

vlmdomain_info — Display the properties of a VLAN domain

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

vlmdomain_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

Output Parameters

vlmdomain_id

The database identifier (ID) of the VLAN domain.

vlmdomain_name

The name of the VLAN domain.

vlmdomain_start_vlan_id

The VLAN identifier (ID) of the first VLAN in the VLAN domain.

vlmdomain_end_vlan_id

The VLAN identifier (ID) of the last VLAN in the VLAN domain.

vlmdomain_description

The description of the VLAN domain.

vlmdomain_class_name

The name of the class applied to the VLAN domain, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

support_vxlan

The type of virtual network used by the domain, VXLAN (1) or VLAN (0).

vlmdomain_class_parameters

The class parameters applied to the VLAN domain and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

vlmdomain_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **vlmdomain_class_parameters**: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&... .

vlmdomain_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

Name

vlmdomain_count — Count the number of VLAN domains

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

group_vlmdomain_add — Add a VLAN domain to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (vlmdomain_id || vlmdomain_name))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmdomain_name

The name of the VLAN domain.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
|------|--|----------------|-----|

| | | | |
|----------------------|----------|----------------------|-----|
| Default value | new_edit | Can be edited | Yes |
|----------------------|----------|----------------------|-----|

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

group_vlmdomain_delete — Remove a VLAN domain from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (vlmdomain_id || vlmdomain_name))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmdomain_name

The name of the VLAN domain.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

vlm_domain_delete — Delete a VLAN domain

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(vlmdomain_id || vlmdomain_name)

Input Parameters

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_name

The name of the VLAN domain.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 65. VLAN Range

Name

vlm_range_add — Add/Edit a VLAN range

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (vlmrange_name && (vlmdomain_id || vlmdomain_name))
- **Editing:** (vlmrange_id || (vlmrange_name && (vlmdomain_id || vlmdomain_name)))

Input Parameters

vlmrange_id

The database identifier (ID) of the VLAN range, a unique numeric key value automatically incremented when you add a VLAN range. Use the ID to specify the VLAN range of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmrange_name

The name of the VLAN range, each VLAN range must have a unique name.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_name

The name of the VLAN domain.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmrange_description

The description of the VLAN range.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

vlmrange_start_vlan_id

The VLAN identifier (ID) of an existing VLAN you want to set as the first VLAN in the VLAN range.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmrange_end_vlan_id

The VLAN identifier (ID) of an existing VLAN you want to set as the last VLAN in the VLAN range.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmrange_disable_overlapping

The overlapping restriction status of the VLAN range. Set it to 1 to prevent VLAN ID overlapping in the range.

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

vlmrange_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

vlmrange_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&... .

Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmrange_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: `<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&...`. If the inheritance or propagation property is not specified, its default value - *set, propagate* - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

vlmrange_list — List the VLAN ranges

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

vlmdomain_id

The database identifier (ID) of the VLAN domain the object belongs to.

vlmrange_id

The database identifier (ID) of the VLAN range.

vlmdomain_name

The name of the VLAN domain the object belongs to.

vlmdomain_start_vlan_id

The VLAN identifier (ID) of the first VLAN in the VLAN domain the object belongs to.

vlmdomain_end_vlan_id

The VLAN identifier (ID) of the last VLAN in the VLAN domain the object belongs to.

vlmdomain_class_name

The name of the class applied to the VLAN domain the object belongs to, it can be preceded by the class directory.

vlmdomain_description

The description of the VLAN domain the object belongs to.

support_vxlan

The type of virtual network used by the domain the range belongs to, VXLAN (1) or VLAN (0).

vlmrange_name

The name of the VLAN range.

vlmrange_start_vlan_id

The VLAN identifier (ID) of the first VLAN in the VLAN range.

vlmrange_end_vlan_id

The VLAN identifier (ID) of the last VLAN in the VLAN range.

vlmrange_description

The description of the VLAN range.

vlmrange_disable_overlapping

The overlapping restriction status of the VLAN range. 1 indicates that when creating VLANs in the range, their IDs should not overlap.

vlmrange_class_name

The name of the class applied to the VLAN range, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

vlmrange_class_parameters

The class parameters applied to the VLAN range and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

vlmrange_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
vlmrange_class_parameters: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&.... .

vlmrange_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

vlmdomain_class_parameters

The class parameters applied to the VLAN domain the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

vlmdomain_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by
vlmdomain_class_parameters: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&.... .

Name

vlmrange_info — Display the properties of a VLAN range

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

vlmrange_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

vlmrange_id

The database identifier (ID) of the VLAN range, a unique numeric key value automatically incremented when you add a VLAN range. Use the ID to specify the VLAN range of your choice.

Output Parameters

vlmdomain_id

The database identifier (ID) of the VLAN domain the object belongs to.

vlmrange_id

The database identifier (ID) of the VLAN range.

vlmdomain_name

The name of the VLAN domain the object belongs to.

vlmdomain_start_vlan_id

The VLAN identifier (ID) of the first VLAN in the VLAN domain the object belongs to.

vlmdomain_end_vlan_id

The VLAN identifier (ID) of the last VLAN in the VLAN domain the object belongs to.

vlmdomain_class_name

The name of the class applied to the VLAN domain the object belongs to, it can be preceded by the class directory.

vlmdomain_description

The description of the VLAN domain the object belongs to.

support_vxlan

The type of virtual network used by the domain the range belongs to, VXLAN (1) or VLAN (0).

vlmrange_name

The name of the VLAN range.

vlmrange_start_vlan_id

The VLAN identifier (ID) of the first VLAN in the VLAN range.

vlmrange_end_vlan_id

The VLAN identifier (ID) of the last VLAN in the VLAN range.

vlmrange_description

The description of the VLAN range.

vlmrange_disable_overlapping

The overlapping restriction status of the VLAN range. 1 indicates that when creating VLANs in the range, their IDs should not overlap.

vlmrange_class_name

The name of the class applied to the VLAN range, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
 - 1 indicates the object is enabled and managed.
 - 2 indicates the object is unmanaged, disabled or both depending on the context.
- By default, *row_enabled* is set to 1 when an object is created.

vlmrange_class_parameters

The class parameters applied to the VLAN range and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

vlmrange_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **vlmrange_class_parameters**: <classparam1>=<inheritance>, <propagation>&<classparam2>=<inheritance>, <propagation>&.... .

vlmrange_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

vlmdomain_class_parameters

The class parameters applied to the VLAN domain the object belongs to and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

vlmdomain_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **vlmdomain_class_parameters**: <classparam1>=<inheritance>, <propagation>&<classparam2>=<inheritance>, <propagation>&.... .

Name

vlmrange_count — Count the number of VLAN ranges

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

group_vlmrange_add — Add a VLAN range to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (vlmrange_id || (vlmrange_name && (vlmdomain_id || vlmdomain_name))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_name

The name of the VLAN domain.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmrange_id

The database identifier (ID) of the VLAN range, a unique numeric key value automatically incremented when you add a VLAN range. Use the ID to specify the VLAN range of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|------|-------------|----------------|-----|
|------|-------------|----------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

vlmrange_name

The name of the VLAN range.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|----------------------|--|-----------------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

group_vlmrange_delete — Remove a VLAN range from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (vlmrange_id || (vlmrange_name && (vlmdomain_id || vlmdomain_name))))

Input Parameters

grp_id

The database identifier (ID) of the group of users which resources you are editing, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_name

The name of the VLAN domain.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmrange_id

The database identifier (ID) of the VLAN range, a unique numeric key value automatically incremented when you add a VLAN range. Use the ID to specify the VLAN range of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|------|-------------|----------------|-----|
| | | | |

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

vlmrange_name

The name of the VLAN range.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

vlm_range_delete — Delete a VLAN range

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(vlmrange_id || (vlmrange_name && (vlmdomain_id || vlmdomain_name)))

Input Parameters

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_name

The name of the VLAN domain.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmrange_id

The database identifier (ID) of the VLAN range, a unique numeric key value automatically incremented when you add a VLAN range. Use the ID to specify the VLAN range of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmrange_name

The name of the VLAN range.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 66. VLAN

Name

vlm_vlan_add — Add/Edit a VLAN

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (vlmvlan_vlan_id && (vlmdomain_id || vlmdomain_name))
- **Editing:** (vlmvlan_id || (vlmvlan_vlan_id && (vlmdomain_id || vlmdomain_name)))

Input Parameters

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmdomain_name

The name of the VLAN domain.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

vlmrange_id

The database identifier (ID) of the VLAN range, a unique numeric key value automatically incremented when you add a VLAN range. Use the ID to specify the VLAN range of your choice.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmrange_name

The name of the VLAN range.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

vlmvlan_id

The database identifier (ID) of the VLAN, a unique numeric key value automatically incremented when you add a VLAN. Use the ID to specify the VLAN of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmvlan_vlan_id

The VLAN identifier (ID) of the VLAN, a unique numeric key value within a VLAN domain. Use the ID to specify the VLAN of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmvlan_name

The name of the VLAN, each VLAN must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

vlmvlan_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

vlmvlan_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmvlan_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: **<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&....**. If the inheritance or propagation property is not specified, its default value - *set, propagate* - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

vlmvlan_list — List the VLANs

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

type

The type of the VLAN (*free* or *used*).

vlmdomain_id

The database identifier (ID) of the VLAN domain the object belongs to.

vlmdomain_name

The name of the VLAN domain the object belongs to.

vlmdomain_start_vlan_id

The VLAN identifier (ID) of the first VLAN in the VLAN domain the object belongs to.

vlmdomain_end_vlan_id

The VLAN identifier (ID) of the last VLAN in the VLAN domain the object belongs to.

vlmdomain_class_name

The name of the class applied to the VLAN domain the object belongs to, it can be preceded by the class directory.

vlmdomain_description

The description of the VLAN domain the object belongs to.

support_vxlan

The type of virtual network used by the domain the VLAN belongs to, VXLAN (1) or VLAN (0).

vlmrange_name

The name of the VLAN range the object belongs to.

vlmrange_id

The database identifier (ID) of the VLAN range the object belongs to.

vlmrange_row_enabled

Internal use. Not documented.

vlmrange_start_vlan_id

The VLAN identifier (ID) of the first VLAN in the VLAN range the object belongs to.

vlmrange_end_vlan_id

The VLAN identifier (ID) of the last VLAN in the VLAN range the object belongs to.

vlmrange_class_name

The name of the class applied to the VLAN range the object belongs to, it can be preceded by the class directory.

vlmrange_description

The description of the VLAN range the object belongs to.

vlmvlan_id

The database identifier (ID) of the VLAN.

vlmvlan_name

The name of the VLAN.

vlmvlan_vlan_id

The VLAN identifier (ID) of the VLAN.

vlmvlan_class_name

The name of the class applied to the object, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

free_start_vlan_id

For free VLAN IDs (**type free**), it returns the first VLAN of the range of VLANs that are not assigned yet. The last VLAN in that range is returned in the parameter **free_end_vlan_id**.

free_end_vlan_id

For free VLAN IDs (**type free**), it returns the last VLAN of a range of VLANs that are not assigned yet. The first VLAN in that range is returned in the parameter **free_start_vlan_id**.

vlmvlan_class_parameters

The class parameters of the VLAN.

vlmvlan_class_parameters_properties

The inheritance property and propagation property of the parameters returned by **vlmvlan_class_parameters**.

vlmvlan_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:
<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

vlmrange_class_parameters

The class parameters of the range the VLAN belongs to.

vlmrange_class_parameters_properties

The inheritance property and propagation property of the parameters returned by **vlmrange_class_parameters**.

vlmdomain_class_parameters

The class parameters of the domain the VLAN belongs to.

vlmdomain_class_parameters_properties

The inheritance property and propagation property of the parameters returned by **vlmdomain_main_class_parameters**.

Name

vlmvlan_info — Display the properties of a VLAN

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

vlmvlan_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

vlmvlan_id

The database identifier (ID) of the VLAN, a unique numeric key value automatically incremented when you add a VLAN. Use the ID to specify the VLAN of your choice.

Output Parameters

type

The type of the VLAN (*free* or *used*).

vlmdomain_id

The database identifier (ID) of the VLAN domain the object belongs to.

vlmdomain_name

The name of the VLAN domain the object belongs to.

vlmdomain_start_vlan_id

The VLAN identifier (ID) of the first VLAN in the VLAN domain the object belongs to.

vlmdomain_end_vlan_id

The VLAN identifier (ID) of the last VLAN in the VLAN domain the object belongs to.

vimdomain_class_name

The name of the class applied to the VLAN domain the object belongs to, it can be preceded by the class directory.

vimdomain_description

The description of the VLAN domain the object belongs to.

support_vxlan

The type of virtual network used by the domain the VLAN belongs to, VXLAN (1) or VLAN (0).

vimrange_name

The name of the VLAN range the object belongs to.

vimrange_id

The database identifier (ID) of the VLAN range the object belongs to.

vimrange_row_enabled

Internal use. Not documented.

vimrange_start_vlan_id

The VLAN identifier (ID) of the first VLAN in the VLAN range the object belongs to.

vimrange_end_vlan_id

The VLAN identifier (ID) of the last VLAN in the VLAN range the object belongs to.

vimrange_class_name

The name of the class applied to the VLAN range the object belongs to, it can be preceded by the class directory.

vimrange_description

The description of the VLAN range the object belongs to.

vimvlan_id

The database identifier (ID) of the VLAN.

vimvlan_name

The name of the VLAN.

vimvlan_vlan_id

The VLAN identifier (ID) of the VLAN.

vimvlan_class_name

The name of the class applied to the object, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

free_start_vlan_id

For free VLAN IDs (**type free**), it returns the first VLAN of the range of VLANs that are not assigned yet. The last VLAN in that range is returned in the parameter **free_end_vlan_id**.

free_end_vlan_id

For free VLAN IDs (**type free**), it returns the last VLAN of a range of VLANs that are not assigned yet. The first VLAN in that range is returned in the parameter **free_start_vlan_id**.

vlmvlan_class_parameters

The class parameters of the VLAN.

vlmvlan_class_parameters_properties

The inheritance property and propagation property of the parameters returned by **vlmvlan_class_parameters**.

vlmvlan_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:
<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

vlmrange_class_parameters

The class parameters of the range the VLAN belongs to.

vlmrange_class_parameters_properties

The inheritance property and propagation property of the parameters returned by **vlmrange_class_parameters**.

vlmdomain_class_parameters

The class parameters of the domain the VLAN belongs to.

vlmdomain_class_parameters_properties

The inheritance property and propagation property of the parameters returned by **vlmdomain_class_parameters**.

Name

vlmvlan_count — Count the number of VLANs

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

vlm_vlan_delete — Delete a VLAN

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(vlmvlan_id || (vlmvlan_vlan_id && (vlmdomain_id || vlmdomain_name)))

Input Parameters

vlmdomain_id

The database identifier (ID) of the VLAN domain, a unique numeric key value automatically incremented when you add a VLAN domain. Use the ID to specify the VLAN domain of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmdomain_name

The name of the VLAN domain.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmrange_id

The database identifier (ID) of the VLAN range, a unique numeric key value automatically incremented when you add a VLAN range. Use the ID to specify the VLAN range of your choice.

| Type | Integer >= 0 | Maximum length | N/A |
|---------------|--------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmrange_name

The name of the VLAN range.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmvlan_id

The database identifier (ID) of the VLAN, a unique numeric key value automatically incremented when you add a VLAN. Use the ID to specify the VLAN of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

vlmvlan_vlan_id

The VLAN identifier (ID) of the VLAN, a unique numeric key value within a VLAN domain. Use the ID to specify the VLAN of your choice.

| | | | |
|----------------------|-------------|-----------------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vlmvlan_name

The name of the VLAN.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Part XIII. VRF Services

Table of Contents

| | |
|--------------------------------------|------|
| 67. VRF | 1187 |
| vrf_vrfobject_add | 1188 |
| vrfobject_list | 1191 |
| vrfobject_info | 1193 |
| vrfobject_count | 1195 |
| vrf_vrfobject_delete | 1196 |
| 68. VRF Route Target | 1198 |
| vrf_linkvrfimportexport_add | 1199 |
| link_vrfimportexport_list | 1202 |
| vrf_linkvrfimportexport_delete | 1204 |

Chapter 67. VRF

Name

vrf_vrfobject_add — Add a VRF

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (vrfobject_name)
- **Editing:** (vrfobject_id || vrfobject_name)

Input Parameters

vrfobject_name

The name of the VRF, each VRF must have a unique name.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

vrfobject_rd_id

The Route Distinguisher (RD) Identifier (ID) of the VRF. Following RFC 4364, the RD ID format must respect one of three types:

Table 67.1. RD ID types and formats

| Type | RD ID Format |
|------|--|
| 0 | <integer between 0 and 65535>:<integer between 0 and 4294967296> |
| 1 | <IPv4 address>:<integer between 0 and 65535> |
| 2 | <integer between 0 and 4294967296>:<integer between 0 and 65535> |

Any other format returns an error.

| | | | |
|---------------|---|----------------|-----|
| Type | Regular expression: (([0-9]+) (25[0-5]) (2[0-4][0-9]) (1[0-9][0-9]) ([1-9][0-9]) ([0-9]))(.)(25[0-5]) (2[0-4][0-9]) ([1-9][0-9]) ([0-9]))(3):[0-9]+ | Maximum length | 128 |
| Default value | | Can be edited | Yes |

vrfobject_comment

The description of the VRF.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

vrfobject_id

The database identifier (ID) of the VRF, a unique numeric key value automatically incremented when you add a VRF. Use the ID to specify the VRF of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vrfobject_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

vrfobject_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: *<class-parameter1>=<value1>&<class-parameter2>=<value2>&... .*

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: *<class-parameter1>&<class-parameter2>&... .* Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

vrfobject_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>_class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: *<class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&... .* If the inheritance or propagation property is not specified, its default value - *set, propagate* - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

vrfobject_list — List the VRFs

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter `offset` must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter `limit` must be specified in **lowercase**.

Output Parameters

vrfobject_id

The database identifier (ID) of the VRF.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause **WHERE**.

vrfobject_rd_id

The Route Distinguisher (RD) Identifier (ID) of the VRF. It can be of type 0 (<integer between 0 and 65535>:<integer between 0 and 4294967296>), 1 (<IPv4 address>:<integer between 0 and 65535>) or 2 (<integer between 0 and 4294967296>:<integer between 0 and 65535>).

vrfobject_name

The name of the VRF.

vrfobject_comment

The description of the VRF.

vrfobject_class_name

The name of the class applied to the VRF, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

vrfobject_class_parameters

The class parameters applied to the VRF and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

vrfobject_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **vrfobject_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

vrfobject_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

vrfobject_info — Display the properties of a VRF

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

vrfobject_id

Input Parameters

vrfobject_id

The database identifier (ID) of the VRF, a unique numeric key value automatically incremented when you add a VRF. Use the ID to specify the VRF of your choice.

Output Parameters

vrfobject_id

The database identifier (ID) of the VRF.

vrfobject_rd_id

The Route Distinguisher (RD) Identifier (ID) of the VRF. It can be of type 0 (<integer between 0 and 65535>:<integer between 0 and 4294967296>), 1 (<IPv4 address>:<integer between 0 and 65535>) or 2 (<integer between 0 and 4294967296>:<integer between 0 and 65535>).

vrfobject_name

The name of the VRF.

vrfobject_comment

The description of the VRF.

vrfobject_class_name

The name of the class applied to the VRF, it can be preceded by the class directory.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, **row_enabled** is set to 1 when an object is created.

vrfobject_class_parameters

The class parameters applied to the VRF and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

vrfobject_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **vrfobject_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... .

vrfobject_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:
<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&... .

Name

vrfobject_count — Count the VRFs

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

vrf_vrfobject_delete — Delete a VRF

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(vrfobject_id || vrfobject_name)

Input Parameters

vrfobject_name

The name of the VRF of your choice.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

vrfobject_id

The database identifier (ID) of the VRF, a unique numeric key value automatically incremented when you add a VRF. Use the ID to specify the VRF of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 68. VRF Route Target

Name

vrf_linkvrfimportexport_add — Add a VRF Route Target

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** `((src_id || (src_rd_id || src_name)) && (dest_id || (dest_rd_id || dest_name)))`
- **Editing:** `((src_id || (src_rd_id || src_name)) && (dest_id || (dest_rd_id || dest_name)))`

Input Parameters

src_id

The database identifier (ID) of the VRF used as the source for the Route Target, a unique numeric key value automatically incremented when you add a VRF. Use the ID to specify the VRF of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

src_name

The name of the VRF used as the source for the Route Target.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

src_rd_id

The Route Distinguisher (RD) identifier (ID) of the VRF used as the source for the Route Target.

| Type | Regular expression: <code>(([0-9]+) ((25[0-5]) (2[0-4][0-9]) (1[0-9][0-9]) ([1-9][0-9]) ([0-9])) (.)([25[0-5]) (2[0-4][0-9]) ([1-9][0-9]) ([0-9])) ([0-9])){3}):[0-9]+</code> | Maximum length | 128 |
|---------------|---|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dest_id

The database identifier (ID) of the VRF used as the target for the Route Target, a unique numeric key value automatically incremented when you add a VRF. Use the ID to specify the VRF of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dest_name

The name of the VRF used as the target for the Route Target.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

dest_rd_id

The Route Distinguisher (RD) identifier (ID) of the VRF used as the target for the Route Target.

| | | | |
|---------------|--|----------------|-----|
| Type | Regular expression: (([0-9]+) (25[0-5]) (2[0-4][0-9]) ([1-9][0-9]) ([1-9][0-9]) ([0-9])) (.)([25[0-5]) (2[0-4][0-9]) ([1-9][0-9]) ([1-9][0-9]) ([0-9])))) ([0-9]+) | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

is_import

A way to determine if the target VRF retrieves the routes of the source VRF (1) or not (0).

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

is_export

A way to determine if the source VRF sends its routes to the target VRF (1) or not (0).

| | | | |
|---------------|-----------------|----------------|-----|
| Type | Boolean: 0 1 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

link_vrfimportexport_list — List the VRF Route Targets

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

link_vrfimportexport_id

The database identifier of the VRF Route Target. It is composed of the database identifiers of the target and source VRF it links as follows: <target-VRF-ID>_<source-VRF-ID> .

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

is_import

A way to determine if the target VRF retrieves the routes of the source VRF (1) or not (0).

is_export

A way to determine if the source VRF sends its routes to the target VRF (1) or not (0).

dest_id

The database identifier of the target VRF.

dest_rd_id

The Route Distinguisher (RD) Identifier (ID) of the target VRF.

dest_name

The name of the target VRF.

src_id

The database identifier of the source VRF.

src_rd_id

The Route Distinguisher (RD) Identifier (ID) of the source VRF.

src_name

The name of the source VRF.

Name

vrf_linkvrfimportexport_delete — Delete a VRF Route Target

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((src_id && dest_id) || (src_name && dest_name))

Input Parameters

dest_id

The database identifier (ID) of the VRF used as the target for the Route Target, a unique numeric key value automatically incremented when you add a VRF. Use the ID to specify the VRF of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

src_id

The database identifier (ID) of the VRF used as the source for the Route Target, a unique numeric key value automatically incremented when you add a VRF. Use the ID to specify the VRF of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

dest_name

The name of the VRF used as the target for the Route Target.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

src_name

The name of the VRF used as the source for the Route Target.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Part XIV. Administration Services

Table of Contents

| | |
|------------------------------------|------|
| 69. Services Management | 1208 |
| service_list | 1209 |
| group_service_add | 1211 |
| group_service_delete | 1213 |
| 70. Group | 1215 |
| group_add | 1216 |
| group_list | 1220 |
| group_info | 1223 |
| group_count | 1225 |
| group_delete | 1226 |
| maintainer_group_list | 1227 |
| 71. User | 1229 |
| user_add | 1230 |
| user_info | 1235 |
| user_service_list | 1237 |
| group_service_list | 1239 |
| group_user_add | 1241 |
| group_user_delete | 1243 |
| user_delete | 1245 |
| 72. Custom Data | 1247 |
| custom_db_data_add | 1248 |
| custom_db_data_list | 1251 |
| custom_db_data_info | 1255 |
| custom_db_data_count | 1258 |
| custom_db_data_groupby | 1259 |
| custom_db_data_groupby_count | 1261 |
| custom_db_data_delete | 1263 |

Chapter 69. Services Management

Name

service_list — List the services

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

service_name

The name of the service.

service_comment

The description of the service.

service_default

The service default assignment. *1* indicates that the service can be executed by any group of users by default. That is to say that any group of users created is granted access to the service. *0* indicates that the administrators must grant the users access to the service if they want to execute it. This only applies to services that can be added as group resource.

service_show

The service visibility in the GUI. *1* indicates that the service is listed among the services of the properties page of a group of users. *0* indicates that the service is not visible in the GUI. For instance, all the <*_count> services are set to *0* as the number of items is visible on all the listing pages.

module_name

The name of the module the service applies to, either IPAM (*ip*), DHCP (*dhcp*) DNS (*dns*), Application (*app*), Guardian (*guardian*), NetChange (*iplocator*), Workflow (*workflow*), Device Manager (*host*), VLAN Manager (*vlm*), VRF (*vrf*), SPX (*SPX*), Administration (*system*), Report (*report*) or Rights & Delegation (*access*).

service_usage

The service type, either *service*, *macro* or *internal*. No user can execute or grant access to the *internal* services as they are internal to SOLIDserver and run on their own when needed.

service_has_help

The service help command availability. *1* indicates that you can use the command <*service_name*> *help* to get the list of input and output parameters. *0* indicates that the help command is unavailable for the service.

service_is_auto_listing

Internal use. Not documented.

Name

group_service_add — Add a service to a group resources/Grant a group access to a service

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (service_id || service_name))

Input Parameters

grp_id

The database identifier (ID) of a group of users, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

service_id

The database identifier (ID) of the service. Use the ID to specify the service of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

service_name

The name of the service.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
|---------------|--|----------------|-----|
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Name

group_service_delete — Remove a service from a group resources/Deny a group access to a service

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (service_id || service_name))

Input Parameters

grp_id

The database identifier (ID) of a group of users, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

service_id

The database identifier (ID) of the service. Use the ID to specify the service of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

service_name

The name of the service.

| Type | String | Maximum length | N/A |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Chapter 70. Group

Name

group_add — Add/Edit a group of users

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** grp_name
- **Editing:** (grp_id || grp_name)

Input Parameters

grp_id

The database identifier (ID) of the group of users, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of users of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

grp_description

The description of the group of users.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

grp_category

A way to determine if the group of the user running the service is the *admin* group (*System*).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 64 |
| Default value | N/A | Can be edited | Yes |

src_grp_id

The database identifier (ID) of an existing group of users you want to copy the rights from.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

src_grp_name

The name of an existing group of users you want to copy the rights from.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

parent_grp_id

The database identifier (ID) of an existing group of users you want to set as the parent of the group of users you are adding/editing.

| | | | |
|---------------|--------------|----------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

parent_grp_name

The name of an existing group of users you want to set as the parent of the group of users you are adding/editing.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

grp_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

grp_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

enabled

Deprecated, replaced by **row_enabled**.

row_enabled

The object activation status.

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.
- If set to 2, the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

| | | | |
|---------------|---------|----------------|-----|
| Type | Integer | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

grp_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter **<object>.class_parameters**.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - *set, propagate* - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errormsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

group_list — List the groups a user belongs to

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

grp_id

The database identifier (ID) of a group of users.

grp_name

The name of the group of users.

grp_category

A way to determine if the group of the user running the service is the *admin* group (*System*).

grp_description

The description of the group of users.

grp_class_name

The name of the class applied to the group, it can be preceded by the class directory.

parent_grp_id

The database identifier (ID) of the parent group of users. 0 indicates that group of users has no parent group.

grp_parent_id_path

The path toward the parent group within the database: <parent-group-name>#<group-ID>#<group-name>#<group-ID>. If the group has no parent, it returns <group-name>#<group-ID>.

grp_level

The level of the group of users, where 0 represents the highest level in the users hierarchy

grp_tmp_level

Internal use. Not documented.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

grp_class_parameters

The class parameters applied to the group and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

grp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **grp_class_parameters**: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&.... .

grp_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

parent_grp_class_parameters

The class parameters applied to the parent group and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

parent_grp_class_parameters_properties

The inheritance and/or propagation properties of the class parameters returned in the parameter **parent_grp_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>.

Name

group_info — Display the properties of a group a user belongs to

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

grp_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

NO_PARENT_CLASS_PARAM

A way to exclude the class parameter details of the parent of the object in the output parameter dedicated to its class parameters.

grp_id

The database identifier (ID) of a group of users, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

Output Parameters

grp_id

The database identifier (ID) of a group of users.

grp_name

The name of the group of users.

grp_category

A way to determine if the group of the user running the service is the *admin* group (*System*).

grp_description

The description of the group of users.

grp_class_name

The name of the class applied to the group, it can be preceded by the class directory.

parent_grp_id

The database identifier (ID) of the parent group of users. 0 indicates that group of users has no parent group.

grp_parent_id_path

The path toward the parent group within the database: <parent-group-name>#<group-ID>#<group-name>#<group-ID>. If the group has no parent, it returns <group-name>#<group-ID>.

grp_level

The level of the group of users, where 0 represents the highest level in the users hierarchy

grp_tmp_level

Internal use. Not documented.

row_enabled

The object activation status:

- 0 indicates the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- 1 indicates the object is enabled and managed.
- 2 indicates the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

grp_class_parameters

The class parameters applied to the group and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

grp_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **grp_class_parameters**: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&.... .

grp_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma: <class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

parent_grp_class_parameters

The class parameters applied to the parent group and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

parent_grp_class_parameters_properties

The inheritance and/or propagation properties of the class parameters returned in the parameter **parent_grp_class_parameters**: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>.

Name

group_count — Count the number of groups the user belongs to

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

group_delete — Delete a group of users

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(grp_id || grp_name)

Input Parameters

grp_id

The database identifier (ID) of a group of users, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

maintainer_group_list — List the groups of users used as Maintainer

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

grp_id

The database identifier (ID) of a group of users.

grp_name

The name of the group of users.

Chapter 71. User

Name

user_add — Add/Edit a user

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** `usr_login`
- **Editing:** (`usr_id` || `usr_login`)

Input Parameters

`usr_id`

The database identifier (ID) of the user, a unique numeric key value automatically incremented when you add a user. Use the ID to specify the user of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

`login`

Deprecated, replaced by `usr_login`.

`usr_login`

The login of the user. A local user and a remote user cannot share the same login account. This login cannot be an email address.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

`description`

Deprecated, replaced by `usr_description`.

`usr_description`

The description of the user.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | | Can be edited | Yes |

`fname`

Deprecated, replaced by `usr_fname`.

`usr_fname`

The first name of the user.

| Type | String | Maximum length | 255 |
|------|--------|----------------|-----|
|------|--------|----------------|-----|

| | | | |
|----------------------|--|----------------------|-----|
| Default value | | Can be edited | Yes |
|----------------------|--|----------------------|-----|

Iname

Deprecated, replaced by **usr_Iname**.

usr_Iname

The last name of the user.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 255 |
| Default value | | Can be edited | Yes |

email

Deprecated, replaced by **usr_email**.

usr_email

The email address of the user.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 255 |
| Default value | | Can be edited | Yes |

type

Deprecated, replaced by **usr_type**.

usr_type

The type of the user, either *local* or *remote*.

| | | | |
|----------------------|---|-----------------------|-----|
| Type | Fixed value: local passwd pam rule | Maximum length | N/A |
| Default value | local | Can be edited | Yes |

path

Deprecated, replaced by **usr_path**.

usr_path

A way to redirect the user your are adding/editing toward the web page of your choice when they log in. Specify the URL of your choice.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 255 |
| Default value | | Can be edited | Yes |

password

Deprecated, replaced by **usr_password**.

usr_password

The password of the user.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

www_settings

Deprecated, replaced by **usr_www_settings**.

usr_www_settings

The URL toward which the user should be directed after being authenticated, in URL format.

| | | | |
|-------------|--------|-----------------------|-----|
| Type | String | Maximum length | 255 |
|-------------|--------|-----------------------|-----|

| | | | |
|----------------------|-----|----------------------|-----|
| Default value | N/A | Can be edited | Yes |
|----------------------|-----|----------------------|-----|

maintainer_grp_id

The database identifier (ID) of an existing group of users you want to set as the maintainer of the user you are adding/editing. Members of the maintainer group are allowed to edit the user information and classes. By default, the maintainer group is the *admin* group.

| | | | |
|----------------------|--------------|-----------------------|-----|
| Type | Integer >= 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

maintainer_grp_name

The name of an existing group of users you want to set as the maintainer of the user you are adding/editing. Members of the maintainer group are allowed to edit the user information and classes. By default, the maintainer group is the *admin* group.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

usr_class_name

The name of the class to apply to the object you are adding, editing or looking for. You must specify the class file directory, e.g. *my_directory/my_class.class*. You cannot use the classes *global* and *default*, they are reserved by the system.

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | 128 |
| Default value | | Can be edited | Yes |

usr_class_parameters

The class parameters to apply to the object you are adding/editing. Specify one or several class parameters and their value, both encoded in URL format: <class-parameter1>=<value1>&<class-parameter2>=<value2>&.... .

| | | | |
|----------------------|--------|-----------------------|-----|
| Type | String | Maximum length | N/A |
| Default value | | Can be edited | Yes |

enabled

Deprecated, replaced by **row_enabled**.

row_enabled

The object activation status.

- If set to 0, the object is present in the database but ignored, i.e. it cannot be managed, counted or listed. This status is applied on objects deleted from the GUI.
- If set to 1, the object is enabled and managed.
- If set to 2, the object is unmanaged, disabled or both depending on the context.

By default, *row_enabled* is set to 1 when an object is created.

| | | | |
|----------------------|---------|-----------------------|-----|
| Type | Integer | Maximum length | N/A |
| Default value | 1 | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

class_parameters_to_delete

A list of all the class parameters that you want to delete. The class parameters must be encoded in URL format and separated by a &: <class-parameter1>&<class-parameter2>&.... Any parameter not specified is still applied to the object with its current inheritance and propagation property configuration.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

usr_class_parameters_properties

The object class parameters inheritance property and propagation property, both encoded in URL format. These properties are ignored if the class parameters you specify are not included in the input parameter <object>.class_parameters.

Specify the class parameters name and the value of their inheritance property and/or propagation property, both encoded in URL format. The parameters specified must be separated by a & and the properties by a comma: <class-parameter1>=<inheritance>,<propagation>&<class-parameter2>=<inheritance>&.... If the inheritance or propagation property is not specified, its default value - set, propagate - is used.

The inheritance property can be set to:

- *inherited*: the object inherits the value of the class parameter from its container, it might inherit it from several levels above.
- *set*: the value of the class parameter is set from the object level, no matter the value of this class parameter on higher levels.
- *inherited_or_set*: the value of the class parameter is either *inherited* from the container if they both have the class parameter configured with the same value or *set* if this class parameter was not defined on the container or had a different value.

The propagation property can be set to:

- *restrict*: the value of the class parameter is only used at this level.
- *propagate*: the value of the class parameter is propagated to the lower level(s).

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Example

In the example below, we call the service **user_add** with PHP (cURL) to add a new SOLIDserver user. In our case, this user has to log in with the credentials *sdsuser* as login and *theirpassword* as password, and we specified an email address: *sdsuser@mydomain.tld*.

Example 71.1. Calling the service user_add using PHP

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "https://solid.intranet/rest/user_add?" .

"usr_login=sdsuser&usr_email=sdsuser%40mydomain.tld&usr_password=theirpassword",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "POST",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #:" . $err;
} else {
    echo $response;
}
```

Name

user_info — Display the properties of the user running the service

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Output Parameters

usr_id

The database identifier (ID) of the user.

usr_login

The login of the user.

usr_type

The type of the user, either *local* or *remote*.

usr_path

The URL the user is redirected to when they log in. If empty, there is no redirection.

usr_description

The description of the user.

usr_fname

The first name of the user.

usr_lname

The last name of the user.

usr_email

The email address of the user.

usr_www_settings

The URL toward which the user should be directed after being authenticated, in URL format.

usr_class_name

The name of the class applied to the user, it can be preceded by the class directory.

maintainer_grp_id

The database identifier (ID) of the user maintainer group.

maintainer_grp_name

The name of the user maintainer group.

usr_class_parameters

The class parameters applied to the user and their value: <class-parameter1>=<value1>&<class-parameter2>=<value2>&....

usr_class_parameters_properties

The inheritance property and/or propagation property of the class parameters returned by **usr_class_parameters**: <classparam1>=<inheritance>,<propagation>&<classparam2>=<inheritance>,<propagation>&....

usr_class_parameters_inheritance_source

The container(s) from which the object inherits its class parameters. The parameters are separated by a & and followed by the type and ID of the container, separated by a comma:

<class-parameter1>=real_<container-type>,<container-ID>&<class-parameter2>=real_<container-type>,<container-ID>&.... .

Name

user_service_list — List the services a user can execute

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

module_name

The name of the module the service applies to, either IPAM (*ip*), DHCP (*dhcp*) DNS (*dns*), Application (*app*), Guardian (*guardian*), NetChange (*iplocator*), Workflow (*workflow*), Device Manager (*host*), VLAN Manager (*vlm*), VRF (*vrf*), SPX (SPX), Administration (*system*), Report (*report*) or Rights & Delegation (*access*).

service_name

The name of the service.

link_active

The service activation status. 1 indicates the service is active.

Name

group_service_list — List the services a user can execute according to the groups they belong to

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

grp_name

The name of the group the user belongs to.

module_name

The name of the module the service applies to, either IPAM (*ip*), DHCP (*dhcp*) DNS (*dns*), Application (*app*), Guardian (*guardian*), NetChange (*iplocator*), Workflow (*workflow*), Device Manager (*host*), VLAN Manager (*vlm*), VRF (*vrf*), SPX (*SPX*), Administration (*system*), Report (*report*) or Rights & Delegation (*access*).

service_name

The name of the service.

link_active

The service activation status. 1 indicates the service is active.

Name

group_user_add — Add a user to a group resources

Description

This service allows you to add an object to the resources of a group. You can only add one object to a group resource per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (usr_id || usr_login))

Input Parameters

grp_id

The database identifier (ID) of a group of users, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

usr_id

The database identifier (ID) of the user, a unique numeric key value automatically incremented when you add a user. Use the ID to specify the user of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

login

Deprecated, replaced by **usr_login**.

usr_login

The login of the user.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
|------|--|----------------|-----|
|------|--|----------------|-----|

| | | | |
|----------------------|----------|----------------------|-----|
| Default value | new_edit | Can be edited | Yes |
|----------------------|----------|----------------------|-----|

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|----------------------|---------------------|-----------------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Example

In the example below, we call the service **group_user_add** with Python (Requests) to add a user as resource to a group of users. The user *sdsuser* will benefit from the objects and rights that the group of users *regular* is configured with.

Example 71.2. Calling the service group_user_add using Python

```
import requests

url = "https://solid.intranet/rest/group_user_add"

querystring = {"grp_name": "regular", "usr_login": "sdsuser"}

headers = {
    'x-ipm-username': "aXBtYWRtaW4=",
    'x-ipm-password': "YWRTaW4=",
    'cache-control': "no-cache"
}

response = requests.request("POST", url, headers=headers, params=querystring)

print(response.text)
```

Name

group_user_delete — Remove a user from a group resources

Description

This service allows you to remove an object from a group resources. You can only remove one object from the resources of a group per call.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

((grp_id || grp_name) && (usr_id || usr_login))

Input Parameters

grp_id

The database identifier (ID) of a group of users, a unique numeric key value automatically incremented when you add a group of users. Use the ID to specify the group of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

grp_name

The name of the group of users.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

usr_id

The database identifier (ID) of the user, a unique numeric key value automatically incremented when you add a user. Use the ID to specify the user of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

login

Deprecated, replaced by **usr_login**.

usr_login

The login of the user.

| Type | String | Maximum length | 128 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

Example

In the example below, we call the service **group_user_delete** with PHP (cURL) to remove a user from the resources of a group of users. When the service is executed, the user *sdsuser* can no longer manage the object or perform the services granted to the group *regular*.

Example 71.3. Calling the service group_user_add using PHP

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL =>
"https://solid.intranet/rest/group_user_delete?grp_name=regular&usr_login=sdsuser",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "DELETE",
    CURLOPT_HTTPHEADER => array(
        "cache-control: no-cache",
        "x-ipm-password: YWRtaW4=",
        "x-ipm-username: aXBtYWRtaW4="
    ),
));

$response = curl_exec($curl);
$err = curl_error($curl);

curl_close($curl);

if ($err) {
    echo "cURL Error #: " . $err;
} else {
    echo $response;
}
```

Name

user_delete — Delete a user

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

(usr_id || usr_login)

Input Parameters

usr_id

The database identifier (ID) of the user, a unique numeric key value automatically incremented when you add a user. Use the ID to specify the user of your choice.

| | | | |
|---------------|-------------|----------------|-----|
| Type | Integer > 0 | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

login

Deprecated, replaced by **usr_login**.

usr_login

The login of the user.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 128 |
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns 0. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Chapter 72. Custom Data

Name

custom_db_data_add — Add/Edit a custom database entry

Description

This service allows you to add objects or edit existing ones. A call can only add or edit one object.

- If no identifier is specified, a new object is added.
- If an existing identifier is specified, the value of all the parameters specified in input edits the corresponding objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

- **Addition:** (custom_db_name_id || custom_db_name)
- **Editing:** custom_db_data_id

Input Parameters

custom_db_data_id

The database identifier (ID) of the custom database entry, a unique numeric key value automatically incremented when you add a custom database entry. Use the ID to specify the custom database entry of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

custom_db_name_id

The database identifier (ID) of the custom database, a unique numeric key value automatically incremented when you add a custom database. Use the ID to specify the custom database of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

custom_db_name

The name of the custom database.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

value1

One of the 10 values you can associate with the custom database entry specified in the parameter **custom_db_data_id**.

| Type | String | Maximum length | 255 |
|---------------|--------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

value2

One of the 10 values you can associate with the custom database entry specified in the parameter **custom_db_data_id**.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

value3

One of the 10 values you can associate with the custom database entry specified in the parameter **custom_db_data_id**.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

value4

One of the 10 values you can associate with the custom database entry specified in the parameter **custom_db_data_id**.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

value5

One of the 10 values you can associate with the custom database entry specified in the parameter **custom_db_data_id**.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

value6

One of the 10 values you can associate with the custom database entry specified in the parameter **custom_db_data_id**.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

value7

One of the 10 values you can associate with the custom database entry specified in the parameter **custom_db_data_id**.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

value8

One of the 10 values you can associate with the custom database entry specified in the parameter **custom_db_data_id**.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

value9

One of the 10 values you can associate with the custom database entry specified in the parameter **custom_db_data_id**.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

value10

One of the 10 values you can associate with the custom database entry specified in the parameter **custom_db_data_id**.

| | | | |
|---------------|--------|----------------|-----|
| Type | String | Maximum length | 255 |
| Default value | N/A | Can be edited | Yes |

add_flag

A way to overload your current operation. Flag the object you are adding/editing if you do not want to edit an existing object that matches your input parameters (*new_only*) or if you want to edit an existing object but not create a new one (*edit_only*).

| | | | |
|---------------|--|----------------|-----|
| Type | Fixed value: new_edit new_only edit_only | Maximum length | N/A |
| Default value | new_edit | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| | | | |
|---------------|---------------------|----------------|-----|
| Type | Fixed value: accept | Maximum length | N/A |
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Name

custom_db_data_list — List the custom database entries

Description

This service allows you to list the objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service `*_list`, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: `SELECT=<param1>,<param2>,...`.

If the call includes the clause `WHERE`, all the parameters it contains must be specified in the statement `SELECT`.

If the call includes the clause `ORDERBY`, all the parameters it contains must be specified in the statement `SELECT`.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, ASC is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

custom_db_data_id

The database identifier (ID) of the custom database entry.

custom_db_name_id

The database identifier (ID) of the custom database.

value1

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value2

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value3

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value4

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value5

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value6

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value7

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value8

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value9

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value10

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

descr

The description of the custom database.

name

The name of the custom database.

type

The type of the custom data, either it belongs to the custom DB Vendor available by default (*system*) or it belongs to a custom DB you created (*read-only*).

read_only

The read-only status of the custom database. 1 indicates that the custom database cannot be edited.

label1

The label to be displayed for the parameter **value1**.

label2

The label to be displayed for the parameter **value2**.

label3

The label to be displayed for the parameter **value3**.

label4

The label to be displayed for the parameter **value4**.

label5

The label to be displayed for the parameter **value5**.

label6

The label to be displayed for the parameter **value6**.

label7

The label to be displayed for the parameter **value7**.

label8

The label to be displayed for the parameter **value8**.

label9

The label to be displayed for the parameter **value9**.

label10

The label to be displayed for the parameter **value10**.

Example

In the example below, we call the service **custom_db_data_list** with PowerShell to display the service help. Everything included before the variable \$headers allows you to skip the SSL certificate checks, if you are using a self-signed certificate.

Example 72.1. Calling the service custom_db_data_list using PowerShell

```
add-type @"
    using System.Net;
    using System.Security.Cryptography.X509Certificates;
    public class TrustAllCertsPolicy : ICertificatePolicy {
        public bool CheckValidationResult(
            ServicePoint srvPoint, X509Certificate certificate,
            WebRequest request, int certificateProblem) {
            return true;
        }
    }
"@
[System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Ssl3,
[Net.SecurityProtocolType]::Tls, [Net.SecurityProtocolType]::Tls11,
```

```
[Net.SecurityProtocolType]::Tls12  
  
$headers = @{ "X-IPM-Username" = 'aXBtYWRtaW4='; "X-IPM-Password" = 'YWRtaW4=' }  
$url = "https://solid.intranet/rest/custom_db_data_list"  
$VRFs = Invoke-RestMethod -Headers $headers -Uri $url -Method Get -DisableKeepAlive  
$VRFs | Format-List
```

Name

custom_db_data_info — Display the properties of a custom database

Description

This service allows you to display the properties of an object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

custom_db_data_id

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

custom_db_data_id

The database identifier (ID) of the custom database entry, a unique numeric key value automatically incremented when you add a custom database entry. Use the ID to specify the custom database entry of your choice.

Output Parameters

custom_db_data_id

The database identifier (ID) of the custom database entry.

custom_db_name_id

The database identifier (ID) of the custom database.

value1

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value2

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value3

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value4

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value5

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value6

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value7

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value8

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value9

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

value10

One of the 10 values associated with the custom database entry returned by the parameter **custom_db_data_id**.

descr

The description of the custom database.

name

The name of the custom database.

type

The type of the custom data, either it belongs to the custom DB Vendor available by default (*system*) or it belongs to a custom DB you created (*read-only*).

read_only

Internal use. Not documented.

label1

The label to be displayed for the parameter **value1**.

label2

The label to be displayed for the parameter **value2**.

label3

The label to be displayed for the parameter **value3**.

label4

The label to be displayed for the parameter **value4**.

label5

The label to be displayed for the parameter **value5**.

label6

The label to be displayed for the parameter **value6**.

label7

The label to be displayed for the parameter **value7**.

label8

The label to be displayed for the parameter **value8**.

label9

The label to be displayed for the parameter **value9**.

label10

The label to be displayed for the parameter **value10**.

Name

custom_db_data_count — Count the number of custom database entries

Description

This service allows you to return the number of objects.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service `*_list` of the object in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : `<parameter>='<value>'` or `<parameter> IS NOT NULL`. The clause is case insensitive and must be encoded in URL format.

Output Parameters

total

The total number of objects matching your input parameters.

¹It is no longer possible to use the structure `<object-name>_class_parameters like <value>` directly in the clause `WHERE`.

Name

custom_db_data_groupby — Group custom databases by parameter(s)

Description

This service, like the SQL statement *GROUPBY*, allows you to gather objects using the columns, i.e. the parameters, specified in input. Unlike other services, the result is not a list of objects but an aggregated result.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: *SELECT=<param1>,<param2>,...*.

If the call includes the clause *WHERE*, all the parameters it contains must be specified in the statement *SELECT*.

If the call includes the clause *ORDERBY*, all the parameters it contains must be specified in the statement *SELECT*.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: *count*, *max*, *min*, *sum* or *avg*. The aggregation function syntax is the following: *SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>)* where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement *SELECT* must also be specified in the statement *GROUPBY*.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service **_list* of the object in this clause, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inheritance_source*.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : *<parameter>='<value>' or <parameter> IS NOT NULL*. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement *SELECT* without aggregation function must be specified in the statement *GROUPBY*.

The statement can contain any output parameter of the service **_list*, except the parameters **_class_parameters*, **_class_parameters_properties* and **_class_parameters_inherit-*

¹It is no longer possible to use the structure *<object-name>_class_parameters like <value>* directly in the clause *WHERE*.

ance_source. If you specify several parameters they must be separated by a comma as follows: `GROUPBY=<param1>,<param2>,...`. The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

ORDERBY

A clause that allows you to sort the result. You can include any output parameter of the service in this clause, except the parameters `*_class_parameters`, `*_class_parameters_properties` and `*_class_parameters_inheritance_source`.

To sort the result using class parameters, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Clause ORDERBY](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as follows: `<parameter>='<value>'`. The clause must be encoded in URL format.

You can add the optional keyword **ASC** (ascending) or **DESC** (descending) after each parameter. If not specified, **ASC** is used by default. The order of the parameters specified is set using their value's name or ordinal number. Each parameter value is compared from one row to the next. If all the parameters of the rows are equal, they are returned in an implementation-dependent order.

offset

The number of rows to skip in the service output.

The input parameter *offset* must be specified in **lowercase**.

limit

The maximum number of results to be returned. Depending on the user resources and the database content, it can return less results than the value you have specified.

The input parameter *limit* must be specified in **lowercase**.

Output Parameters

Your parameters

Any parameter specified in the input statement **SELECT** is returned.

Name

custom_db_data_groupby_count — Count the number of custom databases grouped by parameter(s)

Description

This service allows you to display the total number of results of the service *_groupby.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Input Parameters

SELECT

A statement that allows you to specify which column(s), i.e. parameter, is returned by the service. To decide which parameter aggregates the objects returned, use the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source. The clause must be encoded in URL format.

If you specify several parameters they must be separated by a comma as follows: SELECT=<param1>,<param2>,... .

If the call includes the clause WHERE, all the parameters it contains must be specified in the statement SELECT.

If the call includes the clause ORDERBY, all the parameters it contains must be specified in the statement SELECT.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Each parameter can be associated with an SQL aggregation function: count, max, min, sum or avg. The aggregation function syntax is the following: SELECT=<aggr.-funct.>(<param1>),<aggr.-funct.>(<param2>) where the brackets are required.

If no aggregation function is used, the parameter(s) specified in the statement SELECT must also be specified in the statement GROUPBY.

WHERE

A clause that allows you to filter the result. You can include any output parameter of the service *_list of the object in this clause, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

To filter the result using class parameters, you must tag them first¹. For more details, refer to the section [Including Tagged Class Parameters in the Clause WHERE](#).

The parameters and their value must be specified following the operators and syntax of the SQL standard, as in the following examples : <parameter>='<value>' or <parameter> IS NOT NULL. The clause is case insensitive and must be encoded in URL format.

GROUPBY

A statement that allows you to aggregate the objects returned using the parameter(s) of your choice. Any parameter specified in the statement SELECT without aggregation function must be specified in the statement GROUPBY.

The statement can contain any output parameter of the service *_list, except the parameters *_class_parameters, *_class_parameters_properties and *_class_parameters_inheritance_source.

¹It is no longer possible to use the structure <object-name>_class_parameters like <value> directly in the clause WHERE.

ance_source. If you specify several parameters they must be separated by a comma as follows: GROUPBY=<param1>,<param2>,... . The clause must be encoded in URL format.

To include class parameters in the statement, you must tag them first. For more details, refer to the section [Including Tagged Class Parameters in the Statements SELECT and GROUPBY](#).

Output Parameters

total

The total number of objects matching your input parameters.

Name

custom_db_data_delete — Delete a custom database entry

Description

This service allows you to delete an object. A call can only delete one object.

To execute this service, users must be granted the permission to use it. The rows returned to the user running the service depend on the resources granted to the group they belong to.

Mandatory Input Parameters

custom_db_data_id

Input Parameters

custom_db_data_id

The database identifier (ID) of the custom database entry, a unique numeric key value automatically incremented when you add a custom database entry. Use the ID to specify the custom database entry of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

custom_db_name_id

The database identifier (ID) of the custom database, a unique numeric key value automatically incremented when you add a custom database. Use the ID to specify the custom database of your choice.

| Type | Integer > 0 | Maximum length | N/A |
|---------------|-------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

validate_warnings

A way to bypass (*accept*) any enabled rule that would return warning messages. If the service returns an error message, you cannot bypass the enabled rules.

| Type | Fixed value: accept | Maximum length | N/A |
|---------------|---------------------|----------------|-----|
| Default value | N/A | Can be edited | Yes |

Output Parameters

errno

The status returned by the service after its execution. A successful operation returns *0*. If the operation fails it returns another number, detailed in the appendix [Return Codes](#).

errmsg

The message matching the **errno** returned.

severity

The level of severity of the **errno** returned:

- *Error*: the service cannot be executed.
- *Warning*: the service execution can continue but an issue might have occurred.
- *Notice*: the service execution succeeded.

parameters

The input parameter(s) that caused an issue during the service execution.

param_format

The format that should have been used for the input parameter(s).

param_value

The value of the input parameter(s) that caused the error during the service execution.

ret_oid

The database identifier (ID) of the object you added or edited.

Appendix A. IPAM Cheat Sheet

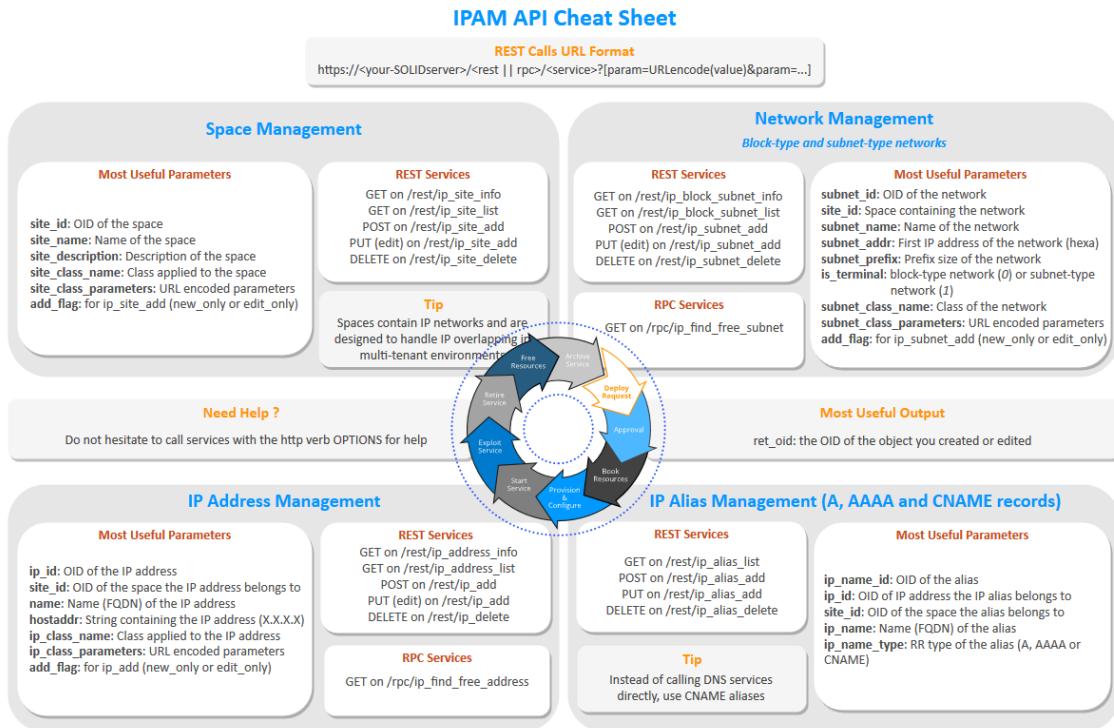


Figure A.1. Some IPAM key services and parameters

Appendix B. IPAM Workflow Sample

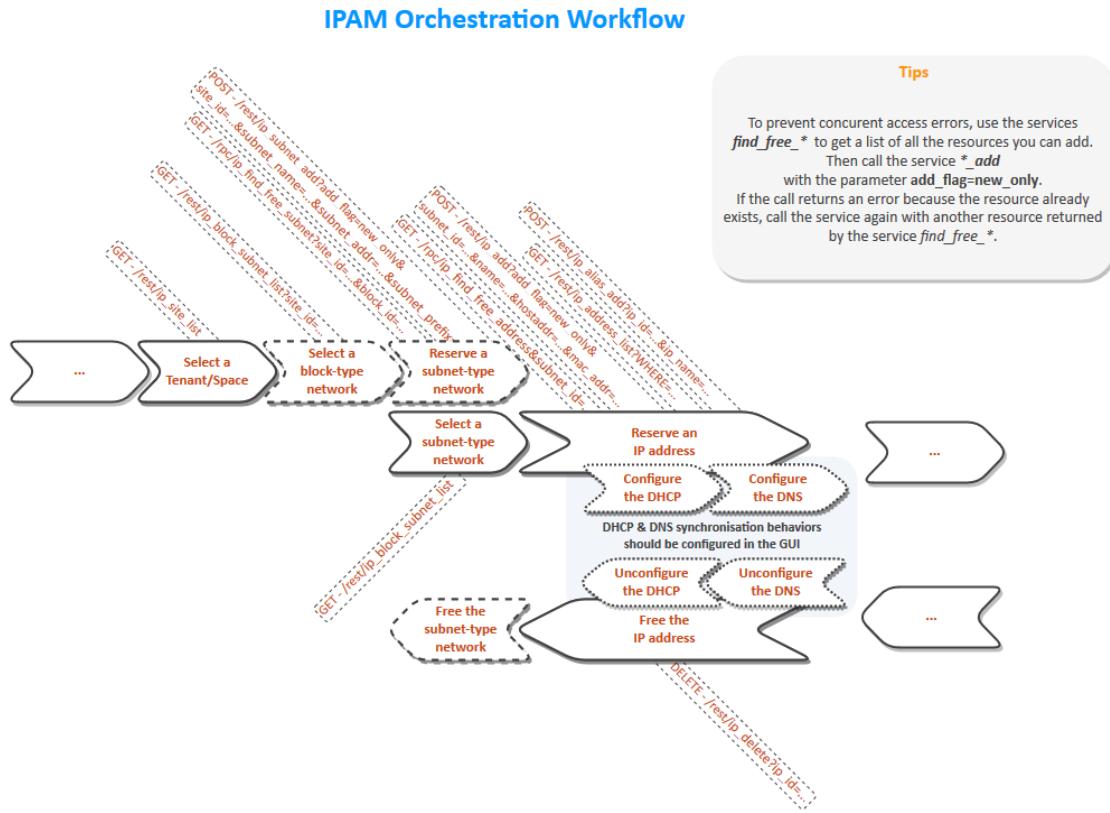


Figure B.1. An IPAM orchestration scenario

Appendix C. DHCP Options

This appendix describes **all the DHCP options** that you can configure **at server, group, scope range and static level**.

They are described following their *Option category* in the GUI configuration wizard:

- [Most Used Options](#).
- [Basic](#).
- [Server Parameters](#).
- [Lease Information](#).
- [WINS/NetBIOS](#).
- [Host IP](#).
- [Interface](#).
- [Servers](#).
- [BootP Compatible](#).
- [DHCP Packet Fields](#).
- [Microsoft DHCP Client](#).
- [NetWare Client](#).
- [NIS/NISplus](#).
- [Miscellaneous](#).
- [Vendor Nwip](#).
- [Vendor MSFT Options](#).

Most Used Options

Table C.1. Most used DHCP options

| Name | Code | Value type | Description |
|----------------------|------|----------------------|---|
| domain-name | 15 | text (name) | The domain name the client uses when resolving name via DNS. |
| domain-name-servers | 6 | list of IP addresses | The list of Domain Name Servers (DNS) available for this client. These servers are listed by order of preference. |
| routers | 3 | list of IP addresses | The list of routers for client subnet. These servers are listed by order of preference. |
| default-lease-time | N/A | duration in seconds | The default lease duration. |
| local-address | N/A | IP address | The IP address that the DHCP server listens on for DHCP requests, rather than listening on all local IP addresses. Configure this option if the server only clients are reached via unicast, such as via DHCP relay agents. |
| max-lease-time | N/A | duration in seconds | The maximum lease duration (unavailable for BOOTP lease). |
| min-lease-time | N/A | duration in seconds | The minimum lease duration. |
| one-lease-per-client | N/A | boolean | Allows you to ensure a server only delivers a single lease to a client. If set to Yes, when a client requests a lease the server frees any other lease the client holds. |

| Name | Code | Value type | Description |
|------------|------|------------|---|
| ping-check | N/A | boolean | Allows you to check via ICMP request if the target address is not used. |

Basic

Table C.2. Basic DHCP options

| Name | Code | Value type | Description |
|---------------------|------|---------------------|--|
| auto-configure | 116 | boolean | Allows you to ask whether, and be notified if, auto-configuration should be disabled on the local subnet. For more details, refer to the RFC2563 available on IETF website at https://tools.ietf.org/html/rfc2563 . |
| broadcast-address | 28 | IP address | The broadcast address for the interface subnet. |
| host-name | 12 | text (name) | The client host name. |
| allow-booting | N/A | boolean | Allows you to decide whether or not to respond to queries from a particular client. It must be set in a host declaration. By default, booting is allowed. If you disable it for a particular client, that client cannot get an address from the DHCP server. |
| allow-bootp | N/A | boolean | Allows you to decide whether or not to respond to <i>bootp</i> queries. By default, <i>bootp</i> queries are allowed. |
| authoritative | N/A | boolean | Allows you to check and allocate leases to clients based on the network segment they connect from. |
| ping-timeout | N/A | duration in seconds | The maximum timeout answer for a ping from the DHCP server. |
| site-option-space | N/A | text (name) | Allows you to specify from which option space the <i>site-local</i> options are taken. The <i>site-local</i> options are the ones with an option code between 224 and 254. |
| use-host-decl-names | N/A | boolean | Allows you to use the hostname of the host declaration as value of the option <i>hostname</i> provided to the clients. |
| vendor-option-space | N/A | boolean | The Option space containing the vendor-specific information. For more details, refer to option 43 (vendor-encapsulated-options). |
| subnet-mask | 1 | IP address | The subnet mask of the connected interface. |

Server Parameters

All server parameter related DHCP options must be preceded by the name of the server they apply to as follows: <server-name>. <option>

Table C.3. Available server parameters

| Name | Code | Value type | Description |
|---------------------------------|------|--------------------------------------|--|
| adaptative-lease-time-threshold | N/A | number between 1 and 100, percentage | The allocated lease time within a range. When the number of allocated leases rises above the percentage stated, all the DHCP clients get the minimal lease time (min-lease-time value). Once |

| Name | Code | Value type | Description |
|-------------------------------|------|---------------------|--|
| | | | the number of allocated leases drops back below the threshold, the server reverts back to normal lease times. |
| authoritative | N/A | boolean | Allows you to check and allocate leases to clients based on the network segment they connect from. |
| leasequery | N/A | boolean | Allows you to specify if the server should respond to DHCLEASEQUERY packets sent by CMTSs, i.e. send back lease information (creation/expiration date...). |
| ping-check | N/A | boolean | Allows you to check via ICMP request if the target address is not used. |
| ping-timeout | N/A | duration in seconds | The maximum timeout answer for a ping from the DHCP server. |
| storm-detection-check-request | N/A | number | <p>The number of requests that have to be received in order to trigger the MAC address black listing. Only MAC addresses associated with an IP address are taken in account in the black list. It means that the clients have to make a DHCP request first with an IP address.</p> <p>This parameter has to be used in conjunction with the parameters <i>storm-detection-check-sec</i> and <i>storm-detection-ignore-sec</i>.</p> <p>If nothing is specified in the field, the default value is 20. If you set a different value, it must be between 1 and 65535.</p> |
| storm-detection-check-sec | N/A | duration in seconds | <p>The period during which the system allows requests. It then checks if it has more than X requests in this time lap, then if it is over, it blacklists the MAC for X seconds.</p> <p>This parameter has to be used in conjunction with the parameters <i>storm-detection-check-request</i> and <i>storm-detection-ignore-sec</i>.</p> <p>If nothing is specified in the field, the default value is 2. If you set a different value, it must be between 1 and 65535.</p> |
| storm-detection-ignore-sec | N/A | duration in seconds | <p>The number of seconds during which any DHCP request from the blacklisted device should be ignored.</p> <p>This parameter has to be used in conjunction with the parameters <i>storm-detection-check-request</i> and <i>storm-detection-check-sec</i>.</p> <p>If nothing is specified in the field, the default value is 30. If you set a different value, it must be between 1 and 65535.</p> |

Lease Information

These options concern the technical mechanisms on the client side of SOLIDserver DHCP protocol.

Table C.4. Lease information options

| Name | Code | Value type | Description |
|---------------------|------|---------------------|--|
| dhcp-rebinding-time | 59 | duration in seconds | The time interval from address assignment until the client transitions to the REBINDING state. |

| Name | Code | Value type | Description |
|-------------------|------|---------------------|---|
| dhcp-renewal-time | 58 | duration in seconds | The time interval from address assignment until the client transitions to the RENEWING state. |

WINS/NetBIOS

Table C.5. WINS/NetBIOS options

| Name | Code | Value type | Description |
|----------------------|------|----------------------|--|
| netbios-dd-server | 45 | list of IP addresses | The list of NetBIOS datagram distribution servers (NBDD), defined by <i>RFC1001</i> and <i>RFC1002</i> . These servers are sorted by order of preference. |
| netbios-name-servers | 44 | list of IP addresses | The list of WINS servers or of Net-BIOS name servers (NBMS). These servers are sorted by order of preference. For more details, refer to <i>RFC1001</i> available on IETF website at https://tools.ietf.org/html/rfc1001 and to <i>RFC1002</i> at https://tools.ietf.org/html/rfc1002 . |
| netbios-node-type | 46 | number | The type of NetBIOS knot described in <i>RFC1001</i> and <i>RFC1002</i> . The value is represented by a numerical code: 1 for B-node, 2 for P-node, 4 for M-node, 8 for H-node. |
| netbios-scope | 47 | text (name) | The Netbios-scope name value of NetBIOS scope specified in <i>RFC1001</i> and <i>RFC1002</i> . |

Host IP

Table C.6. Host IP options

| Name | Code | Value type | Description |
|--------------------------|------|---------------------|---|
| default-ip-ttl | 23 | duration in seconds | The default lifetime that the client must use to send a datagram on the network. Valid values between 1 and 255. |
| ip-forwarding | 19 | boolean | Allows you to enable or disable IP forwarding, accordingly the client should configure its IP layer for packets forwarding. For more details, refer to <i>RFC1533</i> available on IETF website at https://tools.ietf.org/html/rfc1533 . |
| max-dgram-reassembly | 22 | number | The maximum size of datagram which the client must prepare to assemble. |
| non-local-source-routing | 20 | boolean | Allow the source-routing forwarding if the next-hop is on a different physical interface from that crossed by the datagram. For more details, refer to <i>RFC1122</i> available on IETF website at https://tools.ietf.org/html/rfc1122 . |
| path-mtu-aging-timeout | 24 | second | The aging time for the Path MTU Discovery defined for the client. For more details, refer to <i>RFC1191</i> available on IETF website at https://tools.ietf.org/html/rfc1191 . |
| path-mtu-plateau-table | 25 | list of numbers | The list of MTU sizes for the PMTU <i>RFC1191</i> . MTU sizes are prioritized by the order and do not have to be lower than 68. |
| policy-filter | 21 | 2 IP addresses | The filtering policy for the non-local-source-routing. These filters are defined by a list of destination and netmask IP address couples |

| Name | Code | Value type | Description |
|------------------|------|------------|--|
| | | | which specify the destination of entering routes. Any "routedsource" datagram not figuring in the list of filters is destroyed. |
| subnet-selection | 118 | IP address | The DHCP server determines the subnet from which the request originated. For more details, refer to <i>RFC3011</i> available on IETF website at https://tools.ietf.org/html/rfc3011 . |

Interface

Table C.7. Interface options

| Name | Code | Value type | Description |
|-------------------------|------|----------------------|---|
| all-subnets-local | 27 | boolean | Allows you to ensure that all the subnets communicating with the IP interface use the same MTU as the physical interface. |
| arp-cache-timeout | 35 | duration in seconds | The timeout in seconds for ARP cache entries. |
| auto-configure | 116 | boolean | Allows you to ask whether, and be notified if, auto-configuration should be disabled on the local subnet. For more details, refer to the <i>RFC2563</i> available on IETF website at https://tools.ietf.org/html/rfc2563 . |
| broadcast-address | 28 | IP address | The broadcast address for the interface's subnet. |
| classless-static-route | 121 | list of IP addresses | Allows you to use the routers used by the IP protocol to set up a packet transmission path between two IP hosts (one source and one destination host) through the router IP address, listed in the routing table. This option obsoletes the Static Route option (option 33). For more details, refer to the <i>RFC3442</i> available on IETF website at https://tools.ietf.org/html/rfc3442 . |
| default-tcp-ttl | 37 | duration in seconds | The default TTL that the client should use when sending TCP segments. |
| ieee802-3-encapsulation | 36 | boolean | The encapsulation version to use on an Ethernet interface, either Ethernet Version 2 or IEEE 802.3. |
| interface-mtu | 26 | number | The size of MTU to use for this interface, it should be at least 68 bytes. |
| mask-supplier | 30 | boolean | Allows you to specify if the interface must declare its netmask during an ICMP echo. |
| perform-mask-discovery | 29 | boolean | Allows you to specify if the client should attempt an ICMP discovery to find the subnet mask of the local subnet they are connecting to via this interface. Note that using this parameter is not recommended, as the first response received is taken into account and may be incorrect. |
| router-discovery | 31 | boolean | Allows you to specify the client should solicit routers by the "Router Discovery" mechanism when connecting via this interface. For more details, refer to the <i>RFC1256</i> available on IETF website at https://tools.ietf.org/html/rfc1256 . |

| Name | Code | Value type | Description |
|-----------------------------|------|---------------------|---|
| router-solicitation-address | 32 | IP address | The address by which, for this interface, the client must emit its solicitation requests to the routers. |
| static-routes | 33 | 2 IP addresses | In the route interface's cache, the first entry in the list is the destination address and the second is the router's address. The default route (0.0.0.0) is not tolerated here. This option, introduced in <i>RFC2132</i> , was obsoleted by the Classless Static Route Option (option 121). |
| subnet-mask | 1 | IP address | The subnet mask for the network segment to which the client is connected. |
| tcp-keepalive-garbage | 39 | boolean | Allows you to specify if the client must send a garbage byte with a <i>keepalive</i> message. |
| tcp-keepalive-interval | 38 | duration in seconds | The time to wait before sending a keep alive message on a TCP connection. |
| trailer-encapsulation | 34 | boolean | Allows you to specify if the client must negotiate the use of trailers with ARP. For more details, refer to the <i>RFC893</i> available on IETF website at https://tools.ietf.org/html/rfc893 . |

Servers

Table C.8. Server options

| Name | Code | Value type | Description |
|----------------------|------|----------------------|--|
| cookie-servers | 8 | list of IP addresses | The list of cookie servers available for this client. These servers are listed by order of preference. For more details, refer to the <i>RFC865</i> available on IETF website at https://tools.ietf.org/html/rfc865 . |
| finger-server | 73 | list of IP addresses | The list of Finger servers. These servers are sorted by order of preference. |
| font-servers | 48 | list of IP addresses | The list of system-X Windows font servers available for this client. These servers are sorted by order of preference. |
| ien116-name-servers | 5 | list of IP addresses | The list of IEN 116 name servers for this client. These servers must be sorted by preference order. |
| impress-servers | 10 | list of IP addresses | The list of Imagen Impress servers available for this client. These servers are listed by order of preference. |
| irc-server | 74 | list of IP addresses | The list of Internet Relay Chat server. |
| log-servers | 7 | list of IP addresses | The list of UDP log servers (MIT-LCS syslog), available for this client. These servers are listed by order of preference. |
| lpr-servers | 9 | list of IP addresses | The list of printer servers available for this client. These servers are listed by order of preference. For more details, refer to the <i>RFC1179</i> available on IETF website at https://tools.ietf.org/html/rfc1179 . |
| mobile-ip-home-agent | 68 | list of IP addresses | The list of mobile IP home agent. |

| Name | Code | Value type | Description |
|--|------|----------------------|---|
| nis-servers | 41 | list of IP addresses | The list of IP of NIS servers available for the client. The servers can be sorted by order of preference. |
| nisplus-servers | 65 | list of IP addresses | The list of IP addresses of NIS+ servers available for the client. The servers can be sorted by order of preference. |
| nntp-server | 71 | list of IP addresses | The list of NNTP news servers. These servers are sorted by order of preference. |
| ntp-servers | 42 | list of IP addresses | The list of NTP news servers. These servers are sorted by order of preference. |
| pop-server | 70 | list of IP addresses | The list of POP3 message servers. These servers are sorted by order of preference. |
| resource-location-servers | 11 | list of IP addresses | The list of resource servers available for this client. These servers are listed by order of preference. For more details, refer to the <i>RFC887</i> available on IETF website at https://tools.ietf.org/html/rfc887 . |
| leasequery | N/A | boolean | Allows you to specify if the server should respond to DHCLEASEQUERY packets sent by CMTSs, i.e. send back lease information (creation/expiration date...). |
| smtp-server | 69 | list of IP addresses | The list of SMTP message servers. These servers are sorted by order of preference. |
| streettalk-directory-assistance-server | 76 | list of IP addresses | The list of IP addresses in order of preference for STDA servers available to the client. |
| Streettalk-server | 75 | list of IP addresses | The list of StreetTalk servers. These servers are sorted by order of preference. |
| tftp-server-name | 66 | list of IP addresses | The name of the TFTP server to use when the field Sname is used to carry Options. |
| www-server | 72 | list of IP addresses | The list of WEB servers. |
| x-display-manager | 49 | list of IP addresses | The list of X Window XDM system servers. These servers are sorted by order of preference. |

BootP Compatible

Table C.9. BOOTP compatibility options

| Name | Code | Value type | Description |
|---------------------|------|----------------------|---|
| boot-size | 13 | number | The length in block of 512 bytes of the boot image file for this client. |
| bootfile-name | 67 | number | The name of the boot file to use when the File field is used to carry options. |
| cookie-servers | 8 | list of IP addresses | The list of Cookie servers available. These servers are sorted by order of preference. For more details, refer to the <i>RFC865</i> available on IETF website at https://tools.ietf.org/html/rfc865 . |
| domain-name-servers | 6 | list of IP addresses | The list of domain name servers (DNS), available for this client. These servers are listed by order of preference. |

| Name | Code | Value type | Description |
|---------------------------|------|----------------------|---|
| extensions-path | 18 | path | The name of the file containing additional options to be interpreted. The format is described in RFC2132 . For more details, refer to the RFC2132 available on IETF website at https://tools.ietf.org/html/rfc2132 . |
| impress-servers | 10 | list of IP addresses | The list of Imagen Impress servers available for this client. These servers are listed by order of preference. |
| merit-dump | 14 | path | The path of file in which the client must copy the memory image in the event of a crash. This path is constituted by a set of NVT ASCII characters. |
| resource-location-servers | 11 | list of IP addresses | The list of resource servers available for this client. These servers are listed by order of preference. For more details, refer to the RFC887 available on IETF website at https://tools.ietf.org/html/rfc887 . |
| root-path | 17 | path | The path of the disk route for this client. This path is constituted by a set of NVT ASCII characters. |
| filename | N/A | file name | The name of the boot file to use when the field is used to carry options. |
| next-server | N/A | IP address | Allows you to specify the IP address of the server from which the initial boot file (specified in the statement <i>filename</i>) has to be loaded. The <i>server-name</i> should be a numeric IP address. If no next-server parameter applies to a given client, the DHCP server's IP address is used. Some clients prefer to receive the server name in the server-name option. |
| server-name | N/A | text (name) | Allows you to inform the client of the name of the server from which it is booting. This name should be the same as the one provided to the client. |
| swap-server | 16 | IP address | The IP address of the client's Swap Server. |
| time-offset | 2 | duration in seconds | The time offset from UTC (Coordinated Universal Time). |
| time-servers | 4 | IP address | The time server available for this DHCP client. |

DHCP Packet Fields

Table C.10. Packet fields options

| Name | Code | Value type | Description |
|-----------------------------|------|---------------------|--|
| dhcp-client-identifier | 61 | text | A unique identifier for DHCP clients. It allows you to identify clients using the option value rather than their MAC address. |
| dhcp-parameter-request-list | 55 | list of numbers | Allows a DHCP client to request specific option type values from the DHCP server. Each option type is requested and listed by a number value containing a valid or recognized DHCP option code for the server. |
| dhcp-rebinding-time | 59 | duration in seconds | The time interval from address assignment until the client transitions to the REBINDING state. |

| Name | Code | Value type | Description |
|-----------------------------|------|-------------------------------|---|
| dhcp-renewal-time | 58 | duration in seconds | The time interval from address assignment until the client transitions to the RENEWING state. |
| dhcp-server-identifier | 54 | IP address | The identifier is the IP address of the selected server. |
| user-class | 77 | text | The information on the client class. |
| vendor-class-identifier | 60 | text | Allows you to identify the vendor type and configuration of a DHCP client. |
| vendor-encapsulated-options | 43 | <i>provided by the vendor</i> | Allows you to use encapsulated options provided by your vendor. The option can contain a single vendor-specific value or one (or more) vendor-specific sub-options. The configuration set for each vendor is saved in a class. All the vendor classes are kept separately in the DHCP server configuration and trigger a unique response for each vendor. |

Microsoft DHCP Client

Table C.11. Microsoft DHCP client options

| Name | Code | Value type | Description |
|------------------------|------|----------------------|--|
| dhcp-lease-time | 51 | duration in seconds | Allows you to let the clients ask for the lease time of the IP address they are allocated with a request <i>DHCPDISCOVER</i> or <i>DHCPPREQUEST</i> . |
| dhcp-rebinding-time | 59 | duration in seconds | The time interval from address assignment until the client transitions to the REBINDING state. |
| dhcp-renewal-time | 58 | duration in seconds | The time interval from address assignment until the client transitions to the RENEWING state. |
| dhcp-server-identifier | 54 | address | The identifier is the IP address of the selected server. |
| domain-name | 15 | name | The domain name that client should use when resolving hostnames via the Domain Name System. |
| domain-name-servers | 6 | list of IP addresses | A list of Domain Name System name servers available to the client. Servers should be listed in order of preference. |
| domain-search-list | 135 | list of domains | In some circumstances, it is useful for the DHCP client to be configured with the domain search list. Note that Microsoft Windows 200x, XP do not support a list of domain search. |
| netbios-name-servers | 44 | list of IP addresses | The list of Net-BIOS name servers (NBMS). These servers are sorted by order of preference. For more details, refer to <i>RFC1001</i> available on IETF website at https://tools.ietf.org/html/rfc1001 and to <i>RFC1002</i> at https://tools.ietf.org/html/rfc1002 . |
| netbios-node-type | 46 | hexadecimal | The NetBIOS node type option allows NetBIOS over TCP/IP clients which are configurable to be configured as described in <i>RFC1001</i> and <i>RFC1002</i> . Available values are: 0x1 = B-node; 0x2 = P-node; 0x4 = M-node; 0x8 = H-node . |
| netbios-scope | 47 | name | The NetBIOS over TCP/IP scope parameter for the client as specified in <i>RFC1001</i> and <i>RFC1002</i> . |

| Name | Code | Value type | Description |
|------------------|------|----------------------|---|
| routers | 3 | list of IP addresses | A list of IP addresses for routers on the client's subnet. Routers should be listed in order of preference. |
| www-proxy-server | 252 | URL | Allows you to automatically configure proxy settings for the client's browser. Specify the URL of the server that stores the information. |

NetWare Client

Table C.12. NetWare client options

| Name | Code | Value type | Description |
|---------------------|------|-------------------------------|--|
| nds-context | 87 | text | The initial NDS context the client should use. |
| nds-servers | 85 | IP address | One or more NDS servers for the client to contact for access to the NDS database. Servers should be listed in order of preference. |
| nds-tree-name | 86 | name | The initial NDS context the client should use. |
| nwip-domain | 62 | name | Allows you to convey the NetWare/IP domain name used by the NetWare/IP product. |
| autoretries | 8 | <i>provided by the vendor</i> | A list of Quote of the Day servers available to the client. The servers should be listed in order of preference. |
| autoretry-secs | 9 | <i>provided by the vendor</i> | A list of LPR servers available to the client. The servers should be listed in order of preference. |
| nearest-nwip-server | 7 | <i>provided by the vendor</i> | A list of MIT-LCS UDP servers available to the client. The servers should be listed in order of preference. |
| nsq-broadcast | 5 | <i>provided by the vendor</i> | A list of Name servers available to the client. The servers should be listed in order of preference. |
| nwip-1-1 | 10 | <i>provided by the vendor</i> | A list of Imagen Impress servers available to the client. The servers should be listed in order of preference. |
| preferred-dss | 6 | <i>provided by the vendor</i> | A list of DNS servers available to the client. The servers should be listed in order of preference. |
| primary-dss | 11 | <i>provided by the vendor</i> | A list of RLP servers available to the client. The servers should be listed in order of preference. |
| Slp-directory-agent | 78 | address IP | The location of one or more SLP Directory Agents. |
| Slp-service-scope | 79 | scope | The scopes that a SLP Agent is configured to use. |

NIS/NISplus

Table C.13. NIS/NISplus options

| Name | Code | Value type | Description |
|-------------|------|----------------------|---|
| nis-domain | 40 | name | The name of the client's NIS domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set. |
| nis-servers | 41 | list of IP addresses | The list of IP addresses of NIS servers available for the client. The servers can be sorted by order of preference. |

| Name | Code | Value type | Description |
|-----------------|------|----------------------|--|
| nisplus-domain | 64 | name | The name of the client's NIS+ domain. The domain is formatted as a character string consisting of characters from the NVT ASCII character set. |
| nisplus-servers | 65 | list of IP addresses | A list of IP addresses indicating NIS+ servers available to the client. Servers should be listed in order of preference. |

Miscellaneous

Table C.14. Miscellaneous DHCP options

| Name | Code | Value type | Description |
|-----------------------|------|-------------------------------------|---|
| Avaya-96xxx | 242 | ascii string | The private use options - Useful for Avaya 96xxx (Refer to vendor documentation) |
| Mitel-DSCP-Priority | 133 | unsigned integer between 1 and 6 | Allows you to set the IEEE 802.1D/P Layer 2 Priority, useful for Mitel IP phones. For more details, refer to vendor documentation. |
| Mitel-IP-PHONE | 130 | ascii string | Allows you to set a discrimination string, useful for Mitel IP phones. For more details, refer to vendor documentation. |
| Mitel-RTC-Controller | 129 | IP address | Allows you to set the IP address of the call server, useful for Mitel IP phones. For more details, refer to vendor documentation. |
| Mitel-TFTP-Server | 128 | IP address | Allows you to set the IP address of the TFTP server, useful for Mitel IP phones. For more details, refer to vendor documentation. |
| Mitel-VLAN-ID | 132 | unsigned integer between 1 and 4094 | Allows you to set the VLAN ID, useful for Mitel IP phones. For more details, refer to vendor documentation. |
| default-url | 114 | ascii string | The default URL to present in a web browser. For more details, refer to the RFC3679 available on IETF website at https://tools.ietf.org/html/rfc3679 . |
| dhcp-max-message-size | 57 | unsigned integer | Allows a DHCP client or server to specify the maximum size of DHCP message it is willing to accept. The minimum legal value is 576 bytes. For more details, refer to the RFC2132 available on IETF website at https://tools.ietf.org/html/rfc2132 . |
| dhcp-message | 56 | ascii string | Allows a DHCP server to send an error message to a message <i>DHCPNAK</i> to a client in the event of a failure. The option can be used by the client in a message <i>DHCPDECLINE</i> to indicate why the client declined the offered parameters. For more details, refer to the RFC2132 available on IETF website at https://tools.ietf.org/html/rfc2132 . |
| dhcp-message-type | 53 | unsigned integer | Allows you to convey the type of the DHCP message via an integer. Expected values: 1 (<i>DHCPDISCOVER</i>), 2 (<i>DHCPOFFER</i>), 3 (<i>DCHPREQUEST</i>), 4 (<i>DHCPDECLINE</i>), 5 (<i>DHCPACK</i>), 6 (<i>DHCPNAK</i>) or 7 (<i>DHCPRELEASE</i>). For more details, refer to the RFC2132 available on IETF website at https://tools.ietf.org/html/rfc2132 . |

| Name | Code | Value type | Description |
|------------------------|------|--------------------------------------|---|
| dhcp-option-overload | 52 | see RFC | Allows you to indicate that the DHCP fields <i>sname</i> or <i>file</i> are being overloaded when used to carry DHCP options. Specify it if the returned parameters exceed the usual space allotted for options. For more details, refer to the RFC2132 available on IETF website at https://tools.ietf.org/html/rfc2132 . |
| dhcp-requested-address | 50 | IP address | Allows DHCP clients to ask to be assigned a particular IP address in their request (<i>DHCPDIS-COVER</i>). For more details, refer to the RFC2132 available on IETF website at https://tools.ietf.org/html/rfc2132 . |
| domain-search | 119 | list of domains | The DNS domain search list. For more details, refer to the RFC3397 available on IETF website at https://tools.ietf.org/html/rfc3397 . |
| name-service-search | 117 | see RFC | The Name Service Search. For more details, refer to the RFC2937 available on IETF website at https://tools.ietf.org/html/rfc2937 . |
| nwip-suboptions | 63 | see RFC | The NetWare/IP option code will be used to convey all the NetWare/IP related information except for the NetWare/IP domain name. For more details, refer to the RFC2242 available on IETF website at https://tools.ietf.org/html/rfc2242 . |
| always-broadcast | N/A | boolean | Allows you to force the DHCP server to always broadcast its responses to clients by setting this flag to <i>on</i> for the relevant scope. To avoid creating excess broadcast traffic on your network, we recommend limiting this option use to as few clients as possible. |
| always-reply-rfc-1048 | N/A | boolean | Allows the DHCP server to respond with an RFC-1048-style vendor options fields even if the BOOTP clients do not follow the RFC1048 when sending their requests. |
| boot-unknown-clients | N/A | boolean | Allows you to decide whether or not to dynamically assign addresses to unknown clients. Dynamic address assignment to unknown clients is allowed by default. An unknown client is a client that has no host declaration. |
| client-updates | N/A | boolean | Allows you to decide whether or not to honor the client's intention to do its own update of its A record. This is only relevant when doing interim DNS updates. |
| log-facility | N/A | ascii string | Allows the DHCP server to do all of its logging on the specified log facility. |
| min-secs | N/A | unsigned integer between 1 and 255 | The minimum number of seconds since a client began trying to acquire a new lease before the DHCP server responds to its request. |
| remote-port | N/A | unsigned integer between 0 and 65535 | Allows the DHCP server to transmit DHCP responses to DHCP clients upon the UDP port specified in port, rather than on port 68. |

| Name | Code | Value type | Description |
|----------------------|------|------------|---|
| stash-agent-options | N/A | boolean | Allows a server to record the relay agent information options sent to a given client, during the initial <i>DHCPREQUEST</i> message, when the client was in the <i>SELECTING</i> state. The server behaves as if these options were included in all subsequent <i>DHCPREQUEST</i> messages sent in the <i>RENEWING</i> state. |
| update-optimization | N/A | boolean | Allows the server, if set to false, to attempt a DNS update for a client each time they renew their lease, rather than only attempting an update when it appears to be necessary. This will allow the DNS to heal from database inconsistencies more easily, but the cost is that the DHCP server must do many more DNS updates. We recommend enabling this option, its default setting, as it only affects the behavior of the interim DNS update scheme, and has no effect on the ad-hoc DNS update scheme. |
| update-static-leases | N/A | boolean | Allows the DHCP server to do DNS updates for clients even if these clients are being assigned their IP address using a fixed-address statement. This can only work with the interim DNS update scheme. It is not recommended because the DHCP server has no way to tell that the update has been done, and therefore will not delete the record when it is not in use. |
| vivco | 124 | binary | The V-I Vendor Class, or Vendor-Identifying Vendor Class. For more details, refer to the RFC3925 available on IETF website at https://tools.ietf.org/html/rfc3925 . |
| vivso | 125 | binary | The V-I Vendor-Specific Information, or Vendor-Identifying Vendor-Specific Information. For more details, refer to the RFC3925 available on IETF website at https://tools.ietf.org/html/rfc3925 . |

Vendor Nwip

All NetWare/IP Domain Name options below apply to servers, so when configuring these options, make sure to list all the servers in order of preference. For more details, refer to the RFC2242 available on IETF website at <https://tools.ietf.org/html/rfc2242>.

Table C.15. Vendor Nwip options

| Name | Code | Value type | Description |
|---------------------|------|-------------------------------|---|
| autoretries | 8 | <i>provided by the vendor</i> | A list of Quote of the Day servers available to the client. |
| autoretry-secs | 9 | <i>provided by the vendor</i> | A list of LPR servers available to the client. |
| nearest-nwip-server | 7 | <i>provided by the vendor</i> | A list of MIT-LCS UDP servers available to the client. |
| nsq-broadcast | 5 | <i>provided by the vendor</i> | A list of Name servers available to the client. |
| nwip-1-1 | 10 | <i>provided by the vendor</i> | A list of Imagen Impress servers available to the client. |

| Name | Code | Value type | Description |
|---------------|------|-------------------------------|--|
| preferred-dss | 6 | <i>provided by the vendor</i> | A list of DNS servers available to the client. |
| primary-dss | 11 | <i>provided by the vendor</i> | A list of RLP servers available to the client. |

Vendor MSFT

Table C.16. Vendor MSFT options

| Name | Code | Value type | Description |
|---------------------|------|-------------------------------|--|
| default-routers-ttl | 3 | list of IP addresses | A list of 32 bit IP addresses for routers on the client's subnet. The routers should be listed in order of preference. |
| disable-netbios | 1 | <i>provided by the vendor</i> | The subnet mask for the network segment to which the client is connected. |
| release-on-shutdown | 2 | <i>provided by the vendor</i> | The offset of the client's subnet in seconds from Coordinated Universal Time (UTC). |

Appendix D. Return Codes

This section contains the API (Application Programming Interface) return codes for SOLIDserver. For each return code, the following information is provided:

Code

This number corresponds to the number returned by each web service.

Level

Indicates the level that generated the return code. The possible severity codes and their meanings include:

- Error: the web service processing cannot continue.
- Warning: the web service processing can continue, but problems might develop later. You should be cautious.
- Notice: the web service processing can continue, but an issue has been detected.

Description

Explains the circumstances under which this return code might be generated.

Table D.1. SOLIDserver return codes

| Code | Level | Description |
|-------|-------------|---|
| 60000 | Multistatus | 60000: Communications-interrupted. |
| 60001 | Multistatus | 60001: Partner-down. |
| 60002 | Multistatus | 60002: Recovering. |
| 60003 | Multistatus | 60003: Starting up. |
| 60004 | Multistatus | 60004: Server management via SNMP can only be in read-only. We recommend to update to SSL. |
| 60005 | Multistatus | 60005: This server type is no longer supported. To manage a Microsoft server, create a "Microsoft DHCP" server. |
| 60006 | Multistatus | 60006: The shared network is not saved in the DHCP server configuration until it is associated with at least one scope. |
| 60007 | Multistatus | 60007: The shared network is not saved in the DHCP server configuration until it is associated with at least one scope. |
| 60008 | Multistatus | 60008: The prefix delegation is not saved in the DHCP server configuration until it belongs to a shared network associated with at least one scope. |
| 61000 | Multistatus | 61000: Zone type incompatible with Hybrid. |
| 61001 | Multistatus | 61001: Hybrid servers cannot manage authoritative and recursive zones. |
| 61002 | Multistatus | 61002: Hybrid does not support forward on authoritative servers. |
| 61003 | Multistatus | 61003: Hybrid does not support forwarders on authoritative servers. |
| 61004 | Multistatus | 61004: Hybrid server with authoritative zones cannot be recursive. |
| 61005 | Multistatus | 61005: RR type incompatible with Hybrid. |
| 61006 | Multistatus | 61006: Server type incompatible with Hybrid. |
| 61007 | Multistatus | 61007: TSIG keys not supported on Hybrid recursive servers. |
| 61008 | Multistatus | 61008: Hybrid servers do not support views. |
| 61009 | Multistatus | 61009: Hybrid does not support forwarding configuration on authoritative servers. |
| 61010 | Multistatus | 61010: Zone type incompatible with Route 53. |
| 61011 | Multistatus | 61011: RR type incompatible with Route 53. |
| 61012 | Multistatus | 61012: Route 53 servers do not support views. |

| Code | Level | Description |
|-------------|--------------|---|
| 61013 | Multistatus | 61013: At least one character in the value of the record is not supported by Route 53 servers. |
| 61014 | Multistatus | 61014: Only TLD zones are replicated on Route 53 public servers. |
| 61015 | Multistatus | 61015: Maximum number of AWS zones reached. |
| 61016 | Multistatus | 61016: Maximum number of RRset per AWS zone reached. |
| 61017 | Multistatus | 61017: Maximum number of records per RRset per AWS zone reached. |
| 61018 | Multistatus | 61018: This RPZ zone cannot be replicated on one of the physical servers of the smart. |
| 61019 | Multistatus | 61019: The syntax of the BIND include file is incorrect. |
| 61020 | Multistatus | 61020: RRL is not supported on this version of BIND. |
| 61021 | Multistatus | 61021: RRL is not supported on this DNS server. |
| 61022 | Multistatus | 61022: The zone name does not comply with AWS format. |
| 61023 | Multistatus | 61023: The server has no GSS-TSIG key. |
| 61024 | Multistatus | 61024: Server management via SNMP can only be in read-only. We recommend to update to SSL. |
| 61025 | Multistatus | 61025: This server type is no longer supported. To manage a Microsoft server, create a "Microsoft DNS" server. |
| 61026 | Multistatus | 61026: The zone has records configured with geolocation routing policy. You cannot edit or delete it from our GUI. |
| 61027 | Multistatus | 61027: The zone has records configured with routing policy or health check option. You cannot edit or delete it from our GUI. |
| 61028 | Multistatus | 61028: Hybrid servers must be managed via a smart architecture. |
| 61029 | Multistatus | 61029: Maximum number of Azure zones reached. |
| 61030 | Multistatus | 61030: Zone type incompatible with Azure DNS. |
| 61031 | Multistatus | 61031: RR type incompatible with Azure DNS. |
| 61032 | Multistatus | 61032: Maximum number of RRsets per Azure zone reached. |
| 61033 | Multistatus | 61033: Only TLD zones are replicated on Azure DNS servers. |
| 61034 | Multistatus | 61034: Azure DNS servers do not support views. |
| 61035 | Multistatus | 61035: Maximum number of records per RRset per Azure zone reached. |
| 61036 | Multistatus | 61036: The architecture contains views. They are not supported by one of its servers. |
| 61037 | Multistatus | 61037: The architecture contains zone types not supported by one of its servers. |
| 61038 | Multistatus | 61038: The architecture contains RPZ zones. They are not supported by one of its servers. |
| 61039 | Multistatus | 61039: The architecture contains record types not supported by one of its servers. |
| 61040 | Multistatus | 61040: A delegation set of Amazon Route 53 servers cannot have a conflicting domain. |
| 61041 | Multistatus | 61041: TXT record value not supported on Amazon Route 53 servers. |
| 61042 | Multistatus | 61042: Maximum number of private Azure zones reached. |
| 61043 | Multistatus | 61043: Maximum number of RRset per private Azure zone reached. |
| 61044 | Multistatus | 61044: This zone is misconfigured. It belongs to a server configured with at least one VPC already associated with this private domain. |
| 61045 | Multistatus | 61045: Maximum number of zones configured with this delegation set reached. |
| 61046 | Multistatus | 61046: The architecture contains at least one zone that is not a TLD, one of its servers does not support this. |
| 62001 | Multistatus | 62001: A node was disabled due to the health check result. |
| 62010 | Multistatus | 62010: The DNS view does not exist on the Guardian server. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 62011 | Multistatus | 62011: The DNS view is not part of the 16 first ordered view of the Guardian server. |
| 62012 | Multistatus | 62012: At least one of the lists referenced on the rules of the ruleset is invalid. |
| 62013 | Multistatus | 62013: The DNS server, view or zone of the list does not exist. |
| 62014 | Multistatus | 62014: At least one of the rules on the ruleset is invalid. |
| 62015 | Multistatus | 62015: The view of the action Jump to view does not exist on at least one of the Guardian servers of the ruleset. |
| 62016 | Multistatus | 62016: The view of the action Jump to view is not one of the 16 first views of at least one of the Guardian servers of the ruleset. |
| 00004 | Error | Missing parameter(s) [\$parameters]. |
| 00005 | Error | Bad parameter(s) [\$parameters=\$param_value]. |
| 00006 | Error | Permission denied, contact your administrator: you must obtain the relevant rights and/or resources, or ask them to perform the operation for you [\$msg]. |
| 00010 | Error | Parameter '\$parameter' too long (\$max_size max). |
| 00011 | Error | Object '\$obj_name' already exists. |
| 00012 | Error | Bad parameter format (\$parameters) (\$param_value). |
| 00013 | Error | Value cannot be changed (parameter: \$parameter, previous value: \$prev_value, new value: \$new_value). |
| 00014 | Error | Invalid license. |
| 00015 | Error | Internal protocol error (too many values). |
| 00016 | Error | Syntax error at line "\$line" near "\$token". The file "\$file" was not correctly imported. |
| 00018 | Error | Too many connection attempts with invalid credentials. Retry later. |
| 00019 | Error | Please specify some data before performing the search. |
| 00020 | Error | Unauthorized value. This value is dedicated to internal use. |
| 00021 | Error | Unable to set these start and end addresses: invalid range of values. |
| 00100 | Error | Failed to perform your action. (action: \$action, object: \$object). |
| 00101 | Notice | Object added (object: \$object). |
| 00102 | Notice | Object deleted (object: \$object). |
| 00103 | Notice | Object modified (object: \$object). |
| 00104 | Error | Object already exists. |
| 00105 | Error | Cannot modify the parameter. |
| 00106 | Error | This value already exists. |
| 00107 | Error | The operation is impossible on this object (name: \$object_name, type: \$object_type). |
| 00108 | Error | Cannot find object (error 108). |
| 00109 | Error | The same action is already in progress. |
| 00110 | Error | \$message. |
| 00111 | Warning | \$message. |
| 00112 | Notice | \$message. |
| 00200 | Notice | The action was successfully completed. |
| 00201 | Notice | "\$file" was successfully deleted. |
| 00202 | Error | File "\$file" already exists. |
| 00203 | Notice | Class "\$file" copied. |
| 00204 | Error | Unable to edit the class "\$file": it is already used. This class cannot be deleted, disabled, moved, used as a template or no longer used as template for as long as it is applied to a resource. |
| 00205 | Notice | Class "\$file" renamed. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 00206 | Notice | The class "\$file" was successfully duplicated. |
| 00207 | Notice | Class "\$file" has been moved. |
| 00208 | Notice | Initialization completed. |
| 00209 | Error | The class "\$file" already exists. |
| 00210 | Error | Classes located inside 'Library' directory can not be handled. |
| 00211 | Error | This class can not be moved or added to the Library. |
| 00212 | Error | Enabling a space class as a template is not allowed. |
| 00213 | Error | Enabling an IP address class as template is not allowed. |
| 00214 | Error | The backup archive is corrupted or invalid. The restoration process will be aborted. |
| 00215 | Notice | The following key is never used: [\$key]. |
| 00216 | Error | Unable to perform this operation from the Hot Standby: the database is in read-only. Perform it from the Master appliance. |
| 00217 | Error | Unable to export the PDF file: the number of columns is limited to 40. |
| 00219 | Error | Unable to export the data in this format: a PDF file cannot contain more than \$max_pages pages. |
| 00220 | Error | Unable to restore the backup: the selected file belongs to a newer version of SOLIDserver. The restoration process is aborted. |
| 00400 | Error | Could not set an appropriate failover channel to the network \$subnet_name (\$space_name / \$subnet_addr), because \$failovers were found. You will have to update it manually in the networks listing. |
| 00402 | Error | The failover channel "\$dhcpfailover_name" cannot be applied to the network "\$subnet_name" (\$subnet_addr). In the space "\$site_name", this network has already been configured with the failover "\$old_dhcpfailover_name". |
| 00403 | Error | The advanced properties of the DHCP scope "\$dhcpscope" of the server "\$dhcp_name" cannot be replicated to the IPAM: at least one of the existing networks matches it. |
| 00404 | Error | The advanced properties of the type A DNS RR "\$hostaddr" (\$ip_name) cannot be replicated to the IPAM: at least one IP address matches it. |
| 00405 | Error | The advanced properties of the DHCP static "\$hostaddr" linked to the MAC address "\$mac_addr" cannot be replicated to the IPAM: at least one IP address matches it. |
| 00450 | Notice | Migration done. |
| 00501 | Error | SNMP agent timeout: \$hostaddr. |
| 00502 | Error | "SNMP Bad Value" on the device: \$hostaddr. |
| 00503 | Error | "SNMP No such object" on the device: \$hostaddr. |
| 00504 | Error | "SNMP Wrong type" on the device: \$hostaddr. |
| 00505 | Error | "SNMP Generic error" on the device: \$hostaddr. |
| 00506 | Error | "SNMP Authorization error" on the device: \$hostaddr. |
| 00507 | Error | SNMP agent timeout for \$hostaddr: Failure in sendto (Host is down). |
| 02202 | Notice | User enabled (name: \$usr_login). |
| 02203 | Notice | User disabled (name: \$usr_login). |
| 02204 | Notice | Group enabled (name: \$grp_name). |
| 02205 | Notice | Group disabled (name: \$grp_name). |
| 02206 | Notice | Objects add/delete to groups. |
| 02207 | Error | Cannot copy right admin group. |
| 02208 | Error | User without group cannot connect to SOLIDServer, please contact your administrator. |
| 02209 | Error | Group already exist (name: \$grp_name). |
| 02210 | Error | User already exist (login: \$usr_login). |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 02211 | Error | Cannot change parent group: group loop. |
| 02212 | Error | Cannot delete admin group. |
| 02213 | Error | Cannot delete a parent group. |
| 02214 | Error | Cannot find user. |
| 02215 | Error | Cannot delete ipmadmin user. |
| 02216 | Error | Cannot change non local password. |
| 02217 | Error | Cannot find group or user. |
| 02218 | Error | Cannot find group or space. |
| 02219 | Error | Cannot find group or network (block). |
| 02220 | Error | Cannot find group or network (subnet). |
| 02221 | Error | Cannot find group or pool. |
| 02222 | Error | Cannot find group or network (block v6). |
| 02223 | Error | Cannot find group or network (subnet v6). |
| 02224 | Error | Cannot find group or pool v6. |
| 02225 | Error | Cannot find group or DHCP server. |
| 02226 | Error | Cannot find group or DHCP scope. |
| 02227 | Error | Cannot find group or DNS server. |
| 02228 | Error | Cannot find group or DNS view. |
| 02229 | Error | Cannot find group or DNS zone. |
| 02230 | Error | Cannot find group or class. |
| 02231 | Error | Cannot find group or service. |
| 02232 | Error | Only users in the admin group can connect to SOLIDServer. |
| 02234 | Notice | The group "\$grp_name" was successfully added. |
| 02235 | Error | Unable to edit the rights over the class "\$class_name": the class is disabled. |
| 02236 | Error | Unable to edit this class: rights over the classes "default" and "global" cannot be modified. |
| 02237 | Error | Unable to find the group or VLAN domain. |
| 02238 | Error | Unable to find the group or VLAN range. |
| 02239 | Error | Unable to find the group. |
| 02240 | Error | Unable to find the autnum. |
| 02243 | Notice | The right \$service_name was successfully granted to the group \$grp_name. |
| 02244 | Notice | The right \$service_name was successfully denied to the group \$grp_name. |
| 02245 | Error | Unable to change the password: it does not comply with the required password complexity. |
| 02247 | Notice | The API token was successfully added. |
| 02249 | Error | Unable to set the module restrictions: "\$invalid_module" cannot be found. |
| 02250 | Error | Unable to set the module restrictions: you reached the maximum number of restricted modules (\$max_modules). |
| 02251 | Error | Unable to find the API token. |
| 02252 | Notice | The API token was successfully deleted. |
| 02254 | Error | Unable to update the ipmadmin account status: you must use the sds_manage_user command line tool. |
| 02255 | Error | Unable to edit the ipmadmin account: it is disabled. Use the sds_manage_user command line tool instead. |
| 02256 | Error | Unable to edit ipmadmin: the login cannot be changed. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 02257 | Error | Unable to change the password: the user status is No login. An administrator can set a new password using sds_manage_user via CLI to allow this user to log in again. |
| 02259 | Error | Unable to edit the user: the login cannot be ipmadmin. |
| 02300 | Notice | Ping OK (\$hostaddr). |
| 02301 | Error | Ping timeout (\$hostaddr). |
| 02400 | Notice | The rule "\$rule_name" was successfully enabled. |
| 02401 | Notice | The rule "\$rule_name" was successfully disabled. |
| 02402 | Notice | Rule initialized (name: \$rule_name). |
| 02403 | Notice | Rule deleted (name: \$rule_name). |
| 02405 | Error | Cannot initialize disabled rule. |
| 02406 | Error | Unable to add the rule "\$rule_name": this rule name is already used. |
| 02501 | Error | The server did not answer in time to the request for a new scope. |
| 02502 | Error | The server did not answer in time to the request for scope deletion. |
| 02503 | Error | The server did not answer in time to the request for a new option. |
| 02504 | Error | The server didn't answer in time to the request for option deletion. |
| 02505 | Error | The server did not answer in time to the request for a new static. |
| 02506 | Error | The server didn't answer in time to the request for static deletion. |
| 02507 | Error | The server did not answer in time to the request for a new range. |
| 02508 | Error | The server didn't answer in time to the request for range deletion. |
| 02509 | Error | Static already exist (name: \$dhcphost_name, address: \$dhcphost_addr, DHCP server: \$dhcp_name). |
| 02510 | Error | Range overlap [\$range_name]. |
| 02511 | Error | Cannot change DHCP static IP address on a Microsoft DHCP server. |
| 02514 | Error | Can't delete a group with statics. |
| 02515 | Error | Unable to edit the server "\$dhcp_name", it is in read-only: Microsoft DHCP servers with agent are no longer supported. |
| 02520 | Error | You must first select DHCP ranges. |
| 02521 | Error | You must first select DHCP scopes. |
| 02522 | Error | You must first select DHCP leases. |
| 02523 | Notice | DHCP server deleted (name: \$dhcp_name). |
| 02524 | Notice | DHCP group deleted (name: \$dhcpgroup_name). |
| 02525 | Notice | DHCP failover channel deleted (name: \$dhcpfailover_name). |
| 02526 | Notice | DHCP range deleted (name: \$dhcprange_name). |
| 02527 | Notice | DHCP scope deleted (name: \$dhcpscope_name). |
| 02528 | Notice | The DHCP static "\$dhcphost_name" was successfully deleted. |
| 02529 | Notice | DHCP ACL deleted (name: \$dhcpclass_name). |
| 02530 | Notice | The DHCP ACL entry "\$dhcpsubclass_value" was successfully deleted. |
| 02531 | Notice | DHCP lease deleted (name: \$dhcplease_name). |
| 02532 | Notice | Scope modified (name: \$dhcpscope_name, modified parameter: \$parameter). |
| 02533 | Notice | The DHCP scope "\$dhcpscope_name" was successfully copied. |
| 02534 | Notice | The DHCP static "\$dhcphost_name" was successfully copied. |
| 02535 | Notice | The DHCP option "\$dhcption_name" was successfully added to the static. |
| 02536 | Notice | The DHCP option "\$dhcption_name" was successfully deleted from the static. |
| 02537 | Notice | The DHCP scope "\$dhcpscope_name" was successfully moved. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 02538 | Notice | The DHCP scope "\$dhcpscope_name" was successfully added. |
| 02539 | Notice | The DHCP option "\$dhcpoption_name" was successfully added to the scope. |
| 02540 | Notice | The DHCP range "\$dhcprange_name" was successfully added. |
| 02541 | Notice | The DHCP option "\$dhcpoption_name" was successfully added to the range. |
| 02542 | Notice | DHCP range option deleted (name: \$dhcpoption_name). |
| 02543 | Notice | Option \$dhcpoption_name modified (value: \$value). |
| 02544 | Notice | DHCP server synchronized (name: \$dhcp_name). |
| 02545 | Notice | Added group to the static \$dhcpghost_name. |
| 02546 | Notice | The DHCP static "\$dhcpghost_name" was successfully added. |
| 02547 | Notice | Option \$dhcpoption_name modified on static \$dhcpghost_name. |
| 02548 | Notice | Failover channel modified on range \$dhcprange_name. |
| 02549 | Error | The start address (\$new_start_addr) is greater than the end address (\$new_end_addr) on the range "\$dhcprange_name". |
| 02550 | Warning | Could not modify the range \$dhcprange_name: rolling back to the previous range state. |
| 02551 | Notice | Static \$dhcpghost_name modified. |
| 02552 | Notice | The ACL "\$dhcpclass_name" was successfully added. |
| 02553 | Notice | The entry was successfully added on the ACL "\$dhcpclass_name". |
| 02554 | Notice | The option was successfully added on the ACL "\$dhcpclass_name". |
| 02555 | Notice | The option "\$dhcpoption_name" was successfully added to the entry of the ACL "\$dhcpclass_name". |
| 02556 | Notice | The DHCP option "\$dhcpoption_name" was successfully added to the server. |
| 02557 | Notice | The shared network "\$dhcpasn_name" was successfully added. |
| 02558 | Notice | The option definition "\$optiondef_name" was successfully added to the DHCP server "\$dhcp_name". |
| 02559 | Notice | The DHCP option "\$dhcpoption_name" was successfully added to the range. |
| 02560 | Notice | Option \$dhcpoption_name modified (value: \$value). |
| 02561 | Error | ACL not configured for this DHCP server (name: \$dhcpclass_name). |
| 02562 | Error | Shared network not configured for this DHCP server. |
| 02563 | Error | DHCP scope already exists (address: \$dhcpscope_addr, DHCP server: \$dhcp_name). |
| 02564 | Error | DHCP range already exists (address: \$dhcprange_addr, DHCP server: \$dhcp_name). |
| 02565 | Error | Cannot find a DHCP server or a DHCP scope. |
| 02566 | Error | Cannot find DHCP server. |
| 02567 | Error | DHCP ACL already exists (name: \$dhcpclass_name, DHCP server: \$dhcp_name). |
| 02568 | Error | Cannot find a DHCP server or an ACL. |
| 02569 | Error | The DHCP ACL entry already exists (value: \$dhcpsubclass_name, ACL name: \$dhcpclass_name, DHCP server: \$dhcp_name). |
| 02570 | Error | Cannot find DHCP a server or a group. |
| 02571 | Error | DHCP group already exists (name: \$dhcpgroup_name, DHCP server: \$dhcp_name). |
| 02572 | Error | Cannot find the DHCP server of failover channel. |
| 02573 | Error | DHCP failover channel already exists (name: \$dhcpfailover_name, DHCP name: \$dhcp_name, type: \$dhcpfailover_type). |
| 02574 | Error | Can't find DHCP server or shared network. |
| 02575 | Error | Unable to perform the operation: a shared network named "\$dhcpasn_name" already exists on the DHCP server "\$dhcp_name". |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 02576 | Notice | The failover channel "\$dhcpfailover_name" (type: \$dhcpfailover_type) was successfully added to the DHCP server "\$dhcp_name". |
| 02577 | Error | DHCP failover port already used (port: \$port, failover channel name: \$dhcpfailover_name). |
| 02578 | Error | DHCP server already used as secondary for a failover channel (DHCP name: \$dhcp_name). |
| 02579 | Error | Cannot set this DHCP server as secondary: it is already used (DHCP name: \$dhcp_name). |
| 02580 | Error | Can only create shared network on EfficientIP DHCP (DHCP name: \$dhcp_name, type: \$dhcp_type). |
| 02581 | Error | Can't find DHCP server or server option. |
| 02582 | Error | Can't find DHCP scope or scope option. |
| 02583 | Error | Can't find DHCP ACL or ACL option. |
| 02584 | Error | Unable to find the DHCP ACL entry or DHCP option. |
| 02585 | Error | Can't find DHCP group or group option. |
| 02586 | Error | Can't find DHCP range or range option. |
| 02587 | Error | Unable to find the DHCP static or DHCP option. |
| 02588 | Error | Can't find DHCP scope (address: \$dhcpscope_addr, DHCP server: \$dhcp_name). |
| 02590 | Error | Can't find DHCP range (address: \$dhcprange_addr, DHCP server: \$dhcp_name). |
| 02592 | Error | Can't find DHCP static (name: \$dhcpstatic_name, address: \$dhcpstatic_addr, DHCP server: \$dhcp_name). |
| 02594 | Error | Can't find DHCP ACL (name: \$dhcpclass_name, DHCP server: \$dhcp_name). |
| 02596 | Error | Unable to find the DHCP ACL entry (value: \$dhcpsubclass_value, ACL name: \$dhcpclass_name, DHCP server: \$dhcp_name). |
| 02598 | Error | Can't find DHCP group (name: \$dhcpgroup_name, DHCP server: \$dhcp_name). |
| 02600 | Error | Can't find DHCP failover channel (name: \$dhcpfailover_name, DHCP server: \$dhcp_name). |
| 02602 | Error | Can't find DHCP server or option definition. |
| 02603 | Error | DHCP option definition already exists (Name: \$dhcpoptiondef_name, DHCP server: \$dhcp_name). |
| 02604 | Error | Can't modify standard option definition (name: \$dhcpoptiondef_name). |
| 02605 | Error | Can't delete standard option definition (name: \$dhcpoptiondef_name). |
| 02606 | Notice | The option definition "\$dhcpoptiondef_name" was successfully deleted from the DHCP server "\$dhcp_name". |
| 02608 | Error | Can't find DHCP option definition (name: \$dhcpoptiondef_name, DHCP server: \$dhcp_name). |
| 02609 | Error | Can't modify used option definition (name: \$dhcpoptiondef_name). |
| 02610 | Error | Can't delete used option definition (name: \$dhcpoptiondef_name). |
| 02611 | Error | DHCP server name already exists (name: \$dhcp_name). |
| 02612 | Error | Smart DHCP server cannot be used as secondary in failover channel. |
| 02613 | Error | Scope overlap [\$scope_name]. |
| 02614 | Error | DHCP static not in scope (address: \$dhcpstatic_addr, DHCP server: \$dhcp_name). |
| 02615 | Error | Can't find DHCP server, scope or range. |
| 02616 | Error | Can't find DHCP lease. |
| 02617 | Error | Can't delete a children DHCP server. |
| 02618 | Error | Can't delete a SMART DHCP server with children. |
| 02619 | Error | Too many DHCP servers selected. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 02620 | Error | Failover channel not configured for this DHCP server. |
| 02621 | Error | Range overlap [\$dhcprange_name]. |
| 02622 | Error | Too many failover channels set for this smart architecture (DHCP smart server: \$dhcp_name, smart architecture: \$vdhcp_arch). |
| 02623 | Error | Can't delete a DHCP failover currently used by a DHCP scope (Failover name: \$dhcpfailover_name). |
| 02624 | Error | DHCP server address already exists (address: \$hostaddr). |
| 02625 | Error | Can't change DHCP server name (name: \$dhcp_name). |
| 02626 | Error | Can't change DHCP server type (from: \$from, to: \$to). |
| 02627 | Error | Unable to edit the DHCP option "\$optiondef_name": the format of its value is invalid (\$optiondef_value). |
| 02628 | Error | Can't find this DHCP option definition on this server (option name: \$optiondef_name, DHCP server: \$dhcp_name). |
| 02629 | Notice | The DHCP server "\$dhcp_name" was successfully added. |
| 02630 | Notice | The DHCP group "\$dhcpgroup_name" was successfully added. |
| 02631 | Notice | The DHCP lease "\$dhcplease_name" was successfully added. |
| 02632 | Error | DHCP server is in read-only mode (SMART DHCP member). |
| 02633 | Error | Invalid IP addresses (DHCP scope address: \$dhcpscope_addr, DHCP range address: \$dhcprange_addr). |
| 02634 | Error | Specified DHCP scope size is invalid (too small). |
| 02635 | Error | Line \$line could not be parsed. |
| 02636 | Error | Unable to delete the ACL "\$object_name": you cannot delete an already used ACL. |
| 02637 | Notice | DHCP group deleted (name: \$dhcpgroup6_name). |
| 02638 | Notice | DHCP failover channel deleted (name: \$dhcpfailover6_name). |
| 02639 | Notice | The option definition "\$dhcptiondef6_name" was successfully deleted from the DHCP server "\$dhcp6_name". |
| 02640 | Notice | DHCP range deleted. |
| 02641 | Notice | DHCP scope deleted (name: \$dhcpscope6_name). |
| 02642 | Notice | The DHCPv6 static "\$dhcpghost6_name" was successfully deleted. |
| 02643 | Notice | DHCP lease deleted (name: \$dhcplease6_name). |
| 02644 | Error | Your DHCP server is in stateless mode. |
| 02645 | Error | Cannot change DHCP server into stateless mode, you have to delete all ranges and statics first. |
| 02646 | Error | The DHCP static with the IP address \$dhcpghost_addr should be located outside the range \$dhcprange_start_addr - \$dhcprange_end_addr. |
| 02647 | Error | The option \$optiondef_name is not supported on the object \$option_type. |
| 02648 | Notice | The DHCP scope "\$dhcpscope_name" is now Unmanaged. |
| 02649 | Notice | The DHCP scope "\$dhcpscope_name" is now Managed. |
| 02650 | Notice | The DHCP range "\$dhcprange_name" is now Unmanaged. |
| 02651 | Notice | The DHCP range "\$dhcprange_name" is now Managed. |
| 02652 | Notice | The DHCP static "\$dhcpghost_name" is now Unmanaged. |
| 02653 | Notice | The DHCP static "\$dhcpghost_name" is now Managed. |
| 02654 | Notice | The DHCPv6 scope "\$dhcpscope6_name" is now Unmanaged. |
| 02655 | Notice | The DHCPv6 scope "\$dhcpscope6_name" is now Managed. |
| 02656 | Notice | The DHCPv6 range "\$dhcprange6_name" is now Unmanaged. |
| 02657 | Notice | The DHCPv6 range "\$dhcprange6_name" is now Managed. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 02658 | Notice | The DHCPv6 static "\$dhcphost6_name"s now Unmanaged. |
| 02659 | Notice | The DHCPv6 static "\$dhcphost6_name" is now Managed. |
| 02660 | Error | Unable to convert the DHCP physical server into a smart: the MAC address "\$mac-addr" of the static "\$hostname" is invalid. |
| 02663 | Warning | DHCPv6 option definition ignored: \$optiondef_name (\$option_code). |
| 02664 | Notice | The DHCPv6 scope "\$dhcpscope6_name" was successfully added. |
| 02665 | Notice | The DHCPv6 range "\$dhcprange6_name" was successfully added. |
| 02666 | Error | Unable to add/edit the DHCP server "\$dhcp_name": the SNMP parameters cannot be saved. |
| 02667 | Error | Unable to add the option definition "\$dhcptiondef_name". |
| 02669 | Notice | The DHCPv6 group "\$dhcpgroup6_name" was successfully added. |
| 02670 | Notice | The DHCPv6 static "\$dhcphost6_name" was successfully added. |
| 02671 | Error | Unable to add the range: it contains \$range_size addresses (max. size: \$range_max_size). |
| 02672 | Error | Unable to rename the ACL "\$dhcpclass_name" of the server "\$dhcp_name": you cannot edit an ACL name. |
| 02673 | Notice | The option definition "\$optiondef_name" was replaced with a similar option (\$rep_optiondef_name) on the DHCP server "\$dhcp_name". |
| 02674 | Error | Unable to add the option definition "\$optiondef_name" (code:"\$optiondef_code", type:"\$optiondef_type") on the DHCP server "\$dhcp_name": this code is already defined for the option "\$rep_optiondef_name" (code:"\$rep_optiondef_code", type:"\$rep_optiondef_type"). |
| 02676 | Error | Unable to find the DHCP lease. |
| 02677 | Error | The DHCP lease already exists. |
| 02678 | Error | Unable to find the DHCP failover. |
| 02679 | Error | The synchronization of the DHCP server "\$dhcp_name" failed: connection timeout. |
| 02680 | Error | The synchronization of the DHCP server "\$dhcp_name" failed: invalid credentials. |
| 02681 | Notice | The DHCPv6 ACL "\$dhcpclass6_name" was successfully deleted. |
| 02682 | Notice | The DHCPv6 ACL entry "\$dhcpsubclass6_value" was successfully deleted. |
| 02683 | Error | Unable to switch the role of the DHCP server "\$dhcp_name": it cannot become "\$cluster_role". |
| 02684 | Error | Unable to switch the role of the DHCP server "\$dhcp_name" to "\$cluster_role": cluster configuration not found. |
| 02685 | Notice | The role of the DHCP server "\$dhcp_name" was successfully switched to "\$cluster_role" in the cluster. |
| 02686 | Notice | The DHCP cluster "\$dhcp_name" was successfully deleted. |
| 02687 | Error | Unable to delete the DHCP cluster "\$dhcp_name". |
| 02688 | Error | Unable to delete the DHCP cluster "\$dhcp_name": the active DHCP server is unreachable. |
| 02689 | Error | Unable to delete the DHCP cluster "\$dhcp_name": the active DHCP server is unknown. |
| 02690 | Notice | The DHCP cluster was successfully added. |
| 02691 | Error | Unable to add the DHCP cluster. |
| 02692 | Error | Unable to reset the configuration of the passive DHCP server "\$dhcp_name" (error: \$error). |
| 02693 | Notice | The configuration for the passive DHCP server "\$dhcp_name" was successfully reset. |
| 02694 | Error | The DHCP server "\$dhcp_name" is not a DHCP cluster member. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 02695 | Error | The DHCP server "\$dhcp_name" is not a cluster. |
| 02696 | Error | Unable to reset the configuration for the passive DHCP server "\$dhcp_name": you should reset its configuration manually. |
| 02697 | Error | Unable to add "\$dhcp_object_name" (\$dhcp_object_type) to the DHCP cluster "\$dhcp_name": a cluster cannot contain objects. |
| 02698 | Error | "\$dhcp_name" is a cluster container. It must be managed by a smart architecture. |
| 02699 | Error | Unable to convert "\$dhcp_name": DHCP clusters cannot be implicated in a conversion. |
| 02700 | Error | "\$member_name" is a smart architecture. It cannot be managed by the server "\$parent_name". |
| 02701 | Error | "\$dhcp_name" is a physical server. It cannot manage other physical servers (\$vserver_list). |
| 02702 | Error | "\$member_name" is a physical server. It cannot be managed by another physical server (\$parent_name). |
| 02704 | Error | Unable to switch to partner-down a failover in "\$dhcpfailover_state": only the state communication-interrupted can be switched to partner-down. |
| 02705 | Error | Unable to add the static: another one with the same name already exists on this DHCP server. |
| 02706 | Error | Unable to perform this operation: you must synchronize the DNS server first. |
| 02708 | Warning | The DHCPv4 server is already managed by an appliance ("\$serial" at "\$hostaddr") and was last synchronized "\$time" second(s) ago. Click on OK to force management from the new appliance. |
| 02709 | Error | Unable to add/edit the DHCPv4 server ("\$hostaddr"): it is unreachable. |
| 02710 | Error | Unable to add the DHCPv4 server ("\$hostaddr"): it does not match the selected type (EIP DHCPv4 / EIP DHCPv4 Package). |
| 02711 | Error | Unable to add the DHCPv4 server ("\$hostaddr"): the enrollment failed. |
| 02712 | Warning | The DHCPv6 server is already managed by an appliance ("\$serial" at "\$hostaddr") and was last synchronized "\$time" second(s) ago. Click on OK to force management from the new appliance. |
| 02713 | Error | Unable to add/edit the DHCPv6 server ("\$hostaddr"): it is unreachable. |
| 02714 | Error | Unable to add the DHCPv6 server ("\$hostaddr"): it does not match the selected type (EIP DHCPv6 / EIP DHCPv6 Package). |
| 02715 | Error | Unable to add the DHCPv6 server ("\$hostaddr"): the enrollment failed. |
| 02716 | Error | Unable to perform the operation: the prefix "\$prefix" is invalid, it must be between 1 and 128. |
| 02717 | Notice | The prefix delegation "\$dhcpprefix6_name" has been successfully added to the DHCPv6 server "\$dhcp6_name". |
| 02718 | Error | Unable to add the prefix delegation: the prefix is too large for the selected range. |
| 02719 | Error | Unable to add the prefix delegation: it overlaps another prefix delegation. |
| 02721 | Error | Unable to edit the prefix delegation: it cannot be found. |
| 02722 | Error | Unable to add the prefix delegation: it already exists. |
| 02723 | Notice | The prefix delegation has been deleted. |
| 02724 | Error | Unable to add the prefix: it overlaps a DHCP range. |
| 02725 | Error | Unable to add the range: it overlaps another prefix delegation. |
| 02726 | Error | Unable to edit the prefix delegation /\$dhcpprefix6_prefix: the range is too short. It should be large enough to delegate at least one /\$prefix_min. |
| 02727 | Error | Unable to add the prefix delegation: this DHCP server type/architecture does not support this feature. |

| Code | Level | Description |
|-------------|--------------|--|
| 02728 | Error | Unable to edit the type of DHCP server/architecture: at least one prefix delegation is configured. This feature is not supported by the new server/architecture. |
| 02729 | Error | Unable to edit the shared network of the scope (v6): it contains at least one prefix delegation. |
| 02730 | Error | Unable to delete the scope (v6): it is associated to a shared network containing at least one prefix delegation. |
| 02731 | Error | Unable to perform this operation: you cannot edit the DHCP option 43 in the Option space "dhcp". |
| 02732 | Error | Unable to add the selected servers to the smart architecture "\$dhcp_name": Microsoft servers cannot be managed with other server types. |
| 02733 | Error | Unable to add the server to the smart architecture "\$dhcp_name": you cannot manage more than 32 Microsoft servers in one smart architecture. |
| 02734 | Error | Unable to change the role of the servers of the failover channel "\$dhcpfailover_name": you cannot edit the role of Microsoft servers without updating the Peering name. |
| 02735 | Error | Unable to add the option definition \$dhcp_option_code: it already exists in the Option space "\$dhcp_option_space" of the server "\$dhcp_name". |
| 02736 | Notice | The DHCP shared network "\$dhcpsn_name" was successfully deleted. |
| 02737 | Notice | The DHCPv6 shared network "\$dhcpsn6_name" was successfully deleted. |
| 02738 | Error | Unable to delete the shared network "\$dhcpsn_name": it is associated with at least one scope. |
| 02739 | Error | Unable to delete the shared network "\$dhcpsn_name": it is associated with at least one prefix delegation. |
| 02742 | Warning | Unable to import the option "\$option_name" (value: \$option_value) on the "\$option_type": it is not supported. |
| 02743 | Error | Unable to add/edit the DHCP server: a server managed using IPv6 must have a service IP address. |
| 02744 | Error | Unable to add the option definition "\$dhcptiondef_name" on the DHCP server "\$dhcp_name": it cannot contain any parameter Text if Type is array is enabled. |
| 02745 | Error | Unable to add the option definition "\$dhcptiondef_name" on the DHCP server "\$dhcp_name": the parameter Text must be placed last. |
| 02746 | Error | Unable to edit the smart architecture "\$parent_name": you cannot remove the cluster "\$member_name". |
| 02747 | Error | Unable to add the DHCP server to the smart "\$parent_name": "\$member_name" does not support cluster operations. |
| 02748 | Error | Unable to add the cluster "\$dhcp_name": a cluster must contain exactly two servers. |
| 02749 | Error | Unable to add the cluster "\$member_name" to "\$parent_name": a cluster cannot belong to another cluster or server. |
| 02754 | Error | Unable to add the server "\$member_name" to the architecture "\$parent_name": this server is already in a cluster. |
| 02807 | Error | Unable to connect. |
| 02828 | Error | Remote server timeout. |
| 03501 | Error | Access to request \$request_id is denied. |
| 03502 | Error | You're not administrator of request \$request_name. |
| 03509 | Error | This action is not possible on the selected request (\$action \$request_name). |
| 03512 | Error | The request \$request_name can't be archived. |
| 03515 | Error | Unable to archive the request: the specified subdirectory, \$subdirectory, does not exist. |
| 03550 | Notice | The state of the ticket "\$name_ticket" was successfully edited (\$type_ticket). |
| 03600 | Error | You're not allowed to combine or divide an SPX assigned network. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 03602 | Error | It is not allowed to combine SPX assigned networks. |
| 03603 | Notice | Adding send mail reactivate. |
| 03604 | Notice | Deletion send mail reactivate. |
| 03605 | Notice | Confirmation of overflow AW. |
| 03606 | Error | This AutNum already exists. |
| 03607 | Error | Unable to add the IPv6 network: you cannot set an network (inet6num) with a prefix greater than /64 in the RIPE database. |
| 03615 | Error | Unable to add the IPv6 network: the network (inet6num) prefix cannot be greater than the Assignment size prefix in the RIPE database. |
| 03616 | Error | Unable to add the IPv6 network: its prefix does not match the Assignment size set on the parent network (inet6num) in the RIPE database. |
| 03617 | Error | Unable to add the IPv6 network: two levels of network (inet6num) are already set with the status AGGREGATED-BY-LYR in the RIPE database. |
| 03618 | Error | Unable to add the IPv6 network with the status AGGREGATED-BY-LYR: it cannot belong to a network (inet6num) set with the status ASSIGNED in the RIPE database. |
| 03619 | Error | Unable to add the IPv6 network with the status ASSIGNED: it cannot belong to a network (inet6num) set with the status ASSIGNED in the RIPE database. |
| 03620 | Warning | You can set this terminal IPv6 network (inet6num) with the status AGGREGATED-BY-LYR, but terminal networks are usually set with the status ASSIGNED. In the IPAM, a terminal network cannot contain other networks. Do you really want to proceed?. |
| 03621 | Warning | You can remove the class of this IPv6 network, but this does not delete the network (inet6num) from the RIPE database. Do you really want to proceed?. |
| 03622 | Warning | You can edit this IPv6 network, but this change may only apply locally. In the RIPE database, network (inet6num) prefixes greater than /64 are not supported. Do you really want to proceed?. |
| 03623 | Warning | You can edit the status of this IPv6 network, but this change may only apply locally. In the RIPE database, the status of an existing network (inet6num) cannot be edited. Do you really want to proceed?. |
| 03624 | Warning | You can edit the Assignment size of this IPv6 network, but this change may only apply locally. In the RIPE database, the Assignment size of an existing network (inet6num) cannot be edited. Do you really want to proceed?. |
| 03625 | Warning | You can edit the Assignment size of this IPv6 network, but this change may only apply locally. In the RIPE database, the Assignment size prefix cannot be lower than the network (inet6num) prefix. Do you really want to proceed?. |
| 03626 | Warning | You can edit this IPv6 network, but this change may only apply locally. In the RIPE database, the prefix of the network (inet6num) must match the Assignment size set on the parent network (inet6num). Do you really want to proceed?. |
| 03627 | Warning | You can change the status of this IPv6 network to AGGREGATED-BY-LYR, but this change may only apply locally. In the RIPE database, there can only be two levels of networks (inet6num) set with this status. Do you really want to proceed?. |
| 03628 | Warning | You can change the status of this IPv6 network to AGGREGATED-BY-LYR, but this change may only apply locally. In the RIPE database, a network (inet6num) set with this status cannot belong to a network (inet6num) set with the status ASSIGNED. Do you really want to proceed?. |
| 03629 | Warning | You can change the status of this IPv6 network to ASSIGNED, but this change may only apply locally. In the RIPE database, a network (inet6num) set with this status cannot belong to a network (inet6num) set with the status ASSIGNED. Do you really want to proceed?. |
| 03700 | Error | Bookmark already exists. |
| 03701 | Error | You cannot delete a bookmark that you do not own. |
| 03710 | Error | The Smart Folder already exists. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 03711 | Error | You cannot delete a Smart Folder that you do not own. |
| 03720 | Error | Gadget already exists. |
| 03721 | Error | You cannot delete a gadget that you don't own. |
| 03722 | Error | This gadget cannot be deleted. |
| 03723 | Error | You cannot modify a gadget that you don't own. |
| 03724 | Error | This gadget cannot be modified. |
| 03725 | Error | This gadget cannot be Visible to all users. |
| 03726 | Error | Unable to find the gadget and/or user. |
| 03727 | Error | The gadget "\$dashlet_label" has been successfully disabled. |
| 03728 | Error | The Quick Wizard gadget already exists. |
| 03730 | Error | The Quick Wizard already exists. |
| 03731 | Error | You cannot delete a Quick Wizard that you do not own. |
| 03740 | Error | Alert already exists. |
| 03741 | Error | Unable to edit the alert definition: only the user who created it or users of the group "admin" can perform this operation. |
| 03742 | Notice | The alert definition "\$alert_name" was successfully enabled. |
| 03743 | Notice | The alert definition "\$alert_name" was successfully disabled. |
| 03800 | Error | There is no other Active Management server found. Therefore you can not switch this appliance to another state. You first need to add a new Active Management server. |
| 03801 | Error | The selected member(s) could not be deleted because the local management configured in RMAP can not be alone. |
| 04001 | Error | The registry database item does not exist. |
| 04002 | Error | Unable to delete the registry database item: it is in read-only. |
| 05001 | Error | The network "\$start_ip_addr - \$end_ip_addr" has no parent network. |
| 05002 | Error | Unable to add the network "\$subnet_name_to_create" (\$subnet_start_addr_to_create-\$subnet_end_addr_to_create) in the space "\$site_name": it overlaps the network "\$subnet_name" (\$subnet_start_addr-\$subnet_end_addr). |
| 05004 | Error | Can't find a network for that address. |
| 05005 | Error | This address already exists (address: "\$hostaddr", name: "\$ip_name", space: "\$site_name"). |
| 05008 | Error | Pool overlap [\$pool_name]. |
| 05009 | Error | Pool is read only. |
| 05010 | Error | Can't delete default space. |
| 05011 | Error | Network already exists (address: \$subnet_addr, name: \$subnet_name). |
| 05012 | Error | Pool already exists. |
| 05014 | Error | Name already exists as an alias. |
| 05015 | Error | Can't delete IP address in read only pool. |
| 05016 | Error | IP address does not exist. |
| 05017 | Error | Can't delete orphan addresses container. |
| 05018 | Error | Can't delete orphan networks container. |
| 05019 | Error | Can't find a network for this pool. |
| 05020 | Error | Can't delete a space with children. |
| 05021 | Error | Can't delete a space with VLSM networks. |
| 05022 | Error | Invalid network size. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 05023 | Error | The address and the size of the network are not coherent. |
| 05025 | Error | You can't split orphan network. |
| 05026 | Error | An identical IP address already exists in the destination network. |
| 05027 | Error | A network in the target space prevents the migration. |
| 05028 | Error | A pool in the target space prevents the migration. |
| 05029 | Error | A network with the same IP address already exists in the target space. |
| 05030 | Error | Impossible to migrate the network: you cannot migrate orphan networks or addresses. |
| 05031 | Error | A VLSM network can't be migrated. |
| 05032 | Error | Multiple IP addresses match, you must first select a space. |
| 05033 | Error | The specified network does not exist. |
| 05034 | Error | Multiple networks match: you must first select a space. |
| 05036 | Error | You can't split SPX allocated network. |
| 05037 | Error | You can't split a VLSM network. |
| 05038 | Error | You can't split this network, it creates a network overlap. |
| 05040 | Error | You can't migrate VLSM network. |
| 05041 | Error | Unable to migrate the network to another space: it contains at least one non-terminal network. |
| 05042 | Error | The network overlaps another network. |
| 05043 | Error | The target network is not empty, a child network already exists. |
| 05044 | Error | The target network is not empty, a pool already exists. |
| 05045 | Error | The target network is not empty, an address already exists. |
| 05046 | Error | The target network does not exist. |
| 05047 | Error | This address already exists in the target network. |
| 05048 | Error | \$block_name cannot be converted to VLSM (no parent id). |
| 05049 | Error | Critical error: could not query the database. |
| 05050 | Error | Could not find a free IP address. |
| 05051 | Error | You haven't selected a parent VLSM space. |
| 05052 | Error | The selected split value is too big. |
| 05053 | Error | Unable to merge the network "\$subnet_name" (\$start_addr - \$end_addr) in the space "\$site_name". |
| 05054 | Error | Can't split unmanaged networks. |
| 05055 | Error | Unable to merge the network "\$subnet_name" (\$start_addr - \$end_addr) in the space "\$site_name": you cannot merge unmanaged networks. |
| 05056 | Error | Can't delete default space. |
| 05057 | Notice | Generated name: \$name (\$hostaddr). |
| 05058 | Error | Can't change parent space using VLSM networks (blocks). |
| 05059 | Error | Can't change parent space: space loop. |
| 05060 | Error | You must first select networks. |
| 05061 | Error | Space name already exists. |
| 05062 | Error | Not enough free addresses for migration. |
| 05063 | Error | Can't find a space. |
| 05064 | Error | Permission denied to add/modify VLSM network (block). |
| 05066 | Error | Can't delete parent space (name: \$site_name). |
| 05069 | Error | Can't find a space or a network. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 05070 | Error | Can't find network. |
| 05071 | Error | Permission denied to delete VLSM network (block). |
| 05072 | Error | Can't unmanage an orphan networks. |
| 05073 | Error | Unable to find network or pool (\$start_addr - \$end_addr). |
| 05074 | Error | Can't find space or IP address. |
| 05075 | Error | Can't find IP address. |
| 05076 | Error | Can't find IP address or IP address alias. |
| 05077 | Error | IP address alias already exists. |
| 05078 | Error | Can't find IP address alias. |
| 05079 | Error | Permission denied to modify this space. |
| 05080 | Error | Restricted permissions: the rights granted to your group do not allow you to add parent spaces. You can only add child spaces. |
| 05081 | Error | IP address overlaps another IP address in VLSM network (address: \$hostaddr). |
| 05082 | Notice | Network managed. |
| 05083 | Notice | Network unmanaged. |
| 05084 | Error | Cannot unmanage network (block). |
| 05086 | Error | Network already managed. |
| 05087 | Error | Network already unmanaged. |
| 05088 | Error | Unable to unmanage non-terminal networks, no matter their type (block or subnet). |
| 05089 | Error | Cannot create pool in VLSM network. |
| 05090 | Error | Cannot switch to terminal network, you have to delete pools first. |
| 05091 | Error | Cannot switch to VLSM network, you have to delete pools first. |
| 05092 | Error | The network you are creating overlaps assigned IP addresses from two or more lower spaces. |
| 05097 | Error | The network you are creating overlaps assigned IP addresses from the upper space. Please release IP addresses from the upper space before creating your network. |
| 05098 | Error | Can't change network address or size. |
| 05099 | Error | Unable to import the non-terminal network "\$subnet_name" because it can be delegated in more than one VLSM sub-space. Importing imbricated networks requires to have only one sub-space per level or to import networks one level at a time. |
| 05100 | Warning | IP address name already used. DNS and DHCP configurations based on this name can be impacted (Space: \$site_name, Address: \$ip_addr). |
| 05101 | Warning | MAC address already used. (Space: \$site_name, Address: \$ip_addr). |
| 05102 | Error | Permission denied to create object in a template space. |
| 05108 | Error | The specified MAC address doesn't correspond to the chosen Device Manager interface. |
| 05109 | Error | In the \$site_name space, the IP address \$hostaddr has a MAC address. |
| 05110 | Error | In the \$site_name space, the IP address \$hostaddr is not linked to any Device Manager interface. |
| 05111 | Notice | The link between the IP address \$hostaddr (of the \$site_name space) and the Device Manager interface has been successfully removed. |
| 05112 | Notice | The IP address \$hostaddr from the \$site_name space has been successfully linked to the Device Manager interface. |
| 05113 | Error | In the \$site_name space, the IP address \$hostaddr is already linked to an Device Manager interface. |
| 05114 | Error | Unable to merge the selected networks: you cannot merge terminal networks with non-terminal networks. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 05115 | Error | Unable to add the pool: \$pool_addr includes a DHCP static. |
| 05116 | Error | Unable to add the terminal network "\$subnet_name" (\$subnet_start_addr - \$subnet_end_addr): it contains other networks. |
| 05120 | Error | Unable to split the network "\$subnet_name" (\$start_addr-\$end_addr) in the space "\$site_name": you cannot split a network containing pools. |
| 05121 | Error | Unable to replicate the range "\$start_addr - \$end_addr" (DHCP server: \$dhcpserver) to the IPAM: no space can receive it. |
| 05122 | Error | Unable to replicate the scope "\$start_addr - \$end_addr" (DHCP server: \$dhcpserver) to the IPAM: no space can receive it. |
| 05123 | Error | Unable to replicate the static "\$static_ipaddr" (DHCP server: \$dhcpserver) to the IPAM: no space can receive it. |
| 05124 | Error | Restricted permissions: the rights granted to your group do not allow you to delete parent spaces. You can only delete child spaces. |
| 05129 | Error | Unable to add the network "\$subnet_start_addr - \$subnet_end_addr" (\$subnet_name): you cannot allow network reparenting between two spaces. |
| 05130 | Error | Unable to unmanage a network containing pools. |
| 05131 | Error | Unable to split the network "\$subnet_name" (\$start_addr-\$end_addr) in the space "\$site_name": you cannot split a network that overlaps unmanaged networks. |
| 05132 | Error | Unable to migrate to another space: you cannot migrate an unmanaged network. |
| 05135 | Error | Unable to add/edit the pool "\$pool_name" (\$pool_start_addr - \$pool_end_addr): it is configured with DHCP replication but contains IP addresses that are not configured to replicate. |
| 05136 | Error | Your DHCP configuration is invalid: the IP address "\$hostaddr"/"\$name" (space: "\$site_name") does not correspond to any existing lease or static even if it belongs to a pool configured with DHCP replication. |
| 05138 | Error | Unable to edit this parameter: site_is_template cannot be edited. |
| 05139 | Error | Unable to delete the space "\$site_name": it is linked to \$nb_linked_dhcpscope DHCP scope(s). |
| 05140 | Error | Unable to delete the space "\$site_name": it is linked to \$nb_linked_dhcpscope6 DHCPv6 scope(s). |
| 05141 | Error | Unable to delete the space "\$site_name": it is linked to \$nb_linked_dnszone DNS zone(s). |
| 05142 | Error | Unable to delete the space "\$site_name": it is linked to \$nb_linked_ipnetdev NetChange network device(s). |
| 05143 | Error | Unable to find the pool. |
| 05144 | Warning | A CNAME with the same name as the alias "\$alias_name" already exists in the DNS. Do you really want to proceed?. |
| 05145 | Error | Unable to add the terminal network "\$start_ip_addr - \$end_ip_addr": the space "\$site_name" has no block-type network to receive it. |
| 05146 | Error | Unable to add the network "\$subnet_start_addr - \$subnet_end_addr" in the space "\$site_name": its parent network is in read-only because it was retrieved by an Ecosystem rule. |
| 05147 | Error | Unable to add the IP address "\$hostaddr" in the space "\$site_name": its parent network is in read-only because it was retrieved by an Ecosystem rule. |
| 05148 | Error | Unable to unmanage a network retrieved by an Ecosystem rule. |
| 05150 | Error | Unable to add the alias "\$alias_name" (\$alias_type): this name is already used for the IP address. |
| 05151 | Error | Unable to clean the IP address "\$hostaddr" in the space "\$site_name": it is already valid, no cleaning required. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 05152 | Error | The network is too large (/current_prefix): networks cannot have a prefix greater than /8. |
| 05153 | Error | Invalid range (\$subnet6_addr - \$subnet6_end_addr) for the network "\$subnet6_addr/\$subnet6_prefix". |
| 05601 | Error | The VRF "\$vrfobject_name" does not exist. |
| 05606 | Error | Unable to find the VRF Route Target "\$src_name/\$dest_name". |
| 05607 | Error | Unable to find the Target RD of the VRF Route Target (ID: "\$dest_rd_id", name: "\$dest_name"). |
| 05608 | Error | Unable to find the VRF Route Target Source RD (ID: "\$src_rd_id", name: "\$src_name"). |
| 05609 | Error | Unable to edit the VRF Route Target: no Import or Export parameter was set. |
| 05610 | Notice | The VRF was successfully deleted. |
| 05611 | Error | The VRF "\$vrfobject_name" already exists. |
| 05612 | Error | The VRF RD "\$vrfobject_rd_id" already exists. |
| 05613 | Error | Unable to add the VRF Route Target: the Source and Target RD cannot be identical. |
| 05614 | Notice | The VRF "\$vrfobject_name" (\$vrfobject_rd_id) was successfully added. |
| 05615 | Notice | The VRF Route Target "\$src_name/\$dest_name" was successfully added. |
| 05616 | Error | Unable to import the VRF Route Target "\$src_name"/"\$dest_name": it already exists. |
| 05617 | Error | Unable to apply the template class "\$template_name": another object is already using it. |
| 05618 | Error | The specified space does not exist. |
| 05619 | Error | Unable to change the class parameters' inheritance source for the network "\$subnet_name" (IP address: \$subnet_addr; space: \$site_name): this operation can only be performed on VLSM block-type networks. |
| 05620 | Error | Unable to add the IP address: it belongs to a pool replicated in the DHCP, so you must specify a MAC address and tick the box "Create a DHCP static" to add it. |
| 05621 | Error | Unable to edit the IP address name: the box "Use IPAM name instead of DHCP client name" is not ticked. Either tick it again and edit the name or wait for a client to connect and update the IPAM. |
| 05622 | Error | Unable to migrate the network \$subnet_name: it already belongs to the target space \$target_space. |
| 06000 | Error | Unable to find the VLAN domain "\$vlmdomain_name". |
| 06001 | Error | Unable to add the domain "\$vlmdomain_name": it already exists. |
| 06002 | Error | A VLAN domain cannot be reduced. |
| 06003 | Error | Unable to delete the domain: it contains ranges or used VLANs. |
| 06004 | Error | Cannot find the VLAN domain or range. |
| 06005 | Error | Unable to add the range "\$vlmrangename": it already exists. |
| 06006 | Error | Unable to add the range: the VLAN IDs are outside the domain. |
| 06007 | Error | Unable to resize the range: two ranges cannot overlap used VLANs. |
| 06008 | Error | Unable to reduce the size of a range containing used VLANs. |
| 06009 | Error | Unable to find the VLAN Range "\$vlmrangename". |
| 06010 | Error | Unable to delete the range: it contains used VLANs. |
| 06011 | Error | Cannot find VLAN Domain or Vlan. |
| 06012 | Error | Cannot find the VLAN. |
| 06013 | Error | Unable to add the VLAN "\$vlmvlan_vlan_id": it already exists. |
| 06014 | Notice | The domain was successfully deleted. |
| 06015 | Notice | The range was successfully deleted. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 06016 | Notice | The VLAN was successfully deleted. |
| 06017 | Error | Unable to delete the VLAN: it is already Free. |
| 06018 | Error | Unable to add, edit or resize the range: the overlap restriction was set. It overlaps the range "\$vlmrange_name" (ID: \$vlmrange_start_vlan_id - \$vlmrange_end_vlan_id). |
| 06019 | Notice | The VLAN range "\$vlmrange_name" was successfully added. |
| 06020 | Notice | The VLAN domain "\$vlmdomain_name" was successfully added. |
| 06021 | Notice | The VLAN "\$vlmvlan_vlan_id" was successfully added. |
| 06022 | Error | This VLAN ID belongs to several ranges. Please select the "VLAN range" of your choice in the drop-down list. |
| 06023 | Error | This VLAN ID is either used or does not belong to the selected VLAN domain or range. |
| 07500 | Error | Unable to add or edit the authentication rule: the RADIUS server hostname or IP address is invalid. |
| 07501 | Error | Unable to add or edit the authentication rule: the RADIUS server port is invalid. |
| 07502 | Error | Unable to add or edit the authentication rule: the RADIUS server request timeout is invalid. |
| 07503 | Error | Unable to add or edit the authentication rule: the RADIUS server maximum number of tries is invalid. |
| 08000 | Error | No SNMP Agent answered on \$hostaddr. |
| 08001 | Notice | Network device \$hostaddr refreshed (\$name). |
| 08002 | Notice | Network device imported (address: \$hostaddr). |
| 08003 | Error | Invalid/Missing parameter [selected_query/selected_oids]. |
| 08004 | Notice | Network device deleted. |
| 08005 | Notice | Interconnection port forced to yes. |
| 08006 | Notice | Interconnection port autodetected. |
| 08007 | Notice | The port description was successfully edited on the port "\$iplport_name". |
| 08008 | Notice | Port status modified. |
| 08009 | Notice | MAC addresses purged. |
| 08010 | Error | The network device does not exist. |
| 08011 | Error | The network device "\$hostaddr" already exists. |
| 08012 | Error | The network device port already exists. |
| 08014 | Warning | The network device "\$hostaddr" seems to already be present, with a different IP address (\$netdev_addr). |
| 08015 | Error | The service "\$service_name" could not complete: you cannot delete the VLAN "\$iplnetdevvlan_number" from the device "\$iplnetdev_name", it is already used. |
| 08016 | Error | The service "\$service_name" could not complete: you cannot delete the dynamic VLAN "\$iplnetdevvlan_number" from the device "\$iplnetdev_name". |
| 08017 | Error | Unable to perform this operation: the service "\$service_name" is not supported on the device "\$iplnetdev_name". |
| 08018 | Error | The service "\$service_name" failed: the device "\$iplnetdev_name" has returned an SNMP error (\$snmp_error). |
| 08019 | Error | The VLAN "\$vlan_name" (ID: \$vlan_number) is already associated with the port "\$iplport_name" as access/untagged VLAN on the device "\$iplnetdev_name". |
| 08020 | Error | Unable to delete the VLAN "\$vlan_name" (ID: \$vlan_number) from the port "\$iplport_name" on the device "\$iplnetdev_name": you cannot delete access/untagged VLANs. |
| 08021 | Error | The service "\$service_name" could not complete: 802.1X is either disabled or not supported on the device "\$iplnetdev_name". |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 08022 | Notice | The VLAN "\$iplnetdevvlan_name" was successfully deleted. |
| 08023 | Error | The service "\$service_name" could not complete: the SNMP write community is not set on the device "\$iplnetdev_name". |
| 08024 | Error | The service "\$service_name" could not complete: you cannot configure 802.1X authentication if the port "\$iplport_name" is in auto-negotiation on the device "\$iplnetdev_name". |
| 08025 | Error | The service "\$service_name" could not complete: you cannot delete the default VLAN from the device "\$iplnetdev_name". |
| 08026 | Error | The service "\$service_name" could not complete: you cannot configure untagged VLANs on the device "\$iplnetdev_name" because the required MIB is not supported by this device or the port current mode is not compatible. |
| 08027 | Error | The service "\$service_name" could not complete: the port "\$iplport_name" does not support the speed "\$port_speed" on the device "\$iplnetdev_name". |
| 08028 | Error | The service "\$service_name" could not complete: the port "\$iplport_name" does not support the duplex "\$port_duplex" on the device "\$iplnetdev_name". |
| 08029 | Error | This VLAN already exists. |
| 08030 | Error | Unable to edit the VLAN "\$vlan_name": it is reserved by the device "\$iplnetdev_name". |
| 08031 | Error | Port-security configuration is not supported or disabled on the device "\$iplnetdev_name". |
| 08032 | Error | Unable to configure the port "\$iplport_name" on the device "\$iplnetdev_name": you cannot enable both Port-security and 802.1X authentication on a port. |
| 08033 | Notice | The port speed was successfully edited on the port "\$iplport_name". |
| 08034 | Notice | The port duplex mode was successfully edited on the port "\$iplport_name". |
| 08035 | Notice | Port-security was successfully enabled on the port "\$iplport_name". |
| 08036 | Notice | Port-security was successfully disabled on the port "\$iplport_name". |
| 08037 | Notice | 802.1X was successfully enabled on the port "\$iplport_name". |
| 08038 | Notice | 802.1X was successfully disabled on the port "\$iplport_name". |
| 08039 | Notice | The Trunking/Tagging mode was successfully edited on the port "\$iplport_name". |
| 08040 | Notice | The Access/Untagged VLAN was successfully edited on the port "\$iplport_name". |
| 08041 | Notice | The Trunk/Tagged VLAN list was successfully edited on the port "\$iplport_name". |
| 08042 | Error | Unable to edit the Trunking/Tagging mode: the device "\$iplnetdev_name" does not support the mode "\$tagging_mode". |
| 08043 | Notice | The Port-security maximum number of secured MAC addresses was successfully edited on the port "\$iplport_name". |
| 08044 | Error | Unknown VLAN: the VLAN ID "\$vlan_number" cannot be found on the device "\$iplnetdev_name". |
| 08045 | Notice | Port-security mode was successfully modified on the port "\$iplport_name". |
| 08046 | Notice | Port-security action was successfully modified on the port "\$iplport_name". |
| 08047 | Error | The port doesn't exist. |
| 08048 | Notice | The device "\$iplnetdev_name" (\$hostaddr) is already being refreshed. |
| 08049 | Notice | The port "\$iplport_name" was successfully added. |
| 08050 | Notice | The network device "\$iplnetdev_name" was successfully added. |
| 08051 | Notice | The target space of the network device "\$iplnetdev_name" was successfully edited. |
| 08501 | Error | The service "\$service_name" failed: the device "\$iplnetdev_name" has returned an SNMP timeout error. Check your SNMP configuration. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 08502 | Error | The service "\$service_name" could not complete: the device \$iplnetdev_name has returned an SNMP 'bad value error' (on the object "\$obj_name", with the value "\$obj_value"). |
| 08503 | Error | Copy of "running.conf" to "startup.conf": in progress. |
| 08504 | Error | Copy of "running.conf" to "startup.conf": failed. |
| 08505 | Notice | Copy of "running.conf" to "startup.conf": successful. |
| 08506 | Notice | Interconnection status successfully changed: the port \$port (device: \$device) status is set to \$mode. |
| 08507 | Error | Unable to refresh the selected device(s): you did not set any refresh type. |
| 08508 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): the operation failed. |
| 08509 | Error | The configuration file version control is disabled for the device "\$iplnetdev_name" (\$hostaddr). |
| 08510 | Error | Unable to set the connection profile: the device "\$iplnetdev_name" (vendor: \$iplnet-dev_vendor; IP address: \$hostaddr) is not supported. |
| 08511 | Error | Unable to refresh the configuration of the device "\$iplnetdev_name" (vendor: \$iplnet-dev_vendor; IP address: \$hostaddr): it has no connection profile. |
| 08512 | Error | No configuration revision for the device "\$iplnetdev_name" (\$hostaddr). |
| 08513 | Error | Unable to add the configuration revision for the device "\$iplnetdev_name" (\$hostaddr). |
| 08514 | Error | The configuration revision "\$rancid_revision" already exists for the device "\$iplnet-dev_name" (\$hostaddr). |
| 08515 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): invalid password or key. |
| 08516 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): invalid login. |
| 08517 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): failed connection. |
| 08518 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): connection refused. |
| 08519 | Error | Unable to save the configuration files: operation in progress. |
| 08520 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): invalid "enable" password. |
| 08521 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): unsupported method. |
| 08522 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): it does not support versioning. |
| 08523 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): connection failed (timeout). |
| 08524 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): missing password. |
| 08525 | Error | Unable to delete the connection profile "\$rancid_profile_name": it is attached to the network device "\$iplnetdev_name". |
| 08526 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): host key has changed. |
| 08527 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): host unreachable. |
| 08528 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): host key mismatch. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 08529 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): connection closed. |
| 08530 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): missing "enable" password. |
| 08531 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): EOF received. |
| 08532 | Error | Unable to save the configuration file of the device "\$iplnetdev_name" (\$hostaddr): device status is not OK. |
| 08533 | Error | Copy of "running.conf" to "startup.conf": failed (timeout). |
| 08534 | Notice | Snapshot report: [IPv4 address: \$device_ip] [IPv6 address: \$device_ip6] [.pcap file: \$dump_file] [File size: \$filesize MB] [SNMP version: \$snmp_version] [sysObjectID: \$sysobjectid] [sysDescr: \$sysdescr] [Vendor: \$vendor] [Number of packets: \$nb_objects] [Number of ports: \$nb_ports] [Number of VLANs: \$nb_vlans] [Number of MAC addresses discovered: \$discovered_mac] [ARP cache: \$arp]. |
| 08535 | Error | Unable to analyze the device with this SNMP version. |
| 08536 | Error | Incomplete snapshot: missing object (\$obj_name). |
| 08537 | Error | Unable to analyze the device snapshot. |
| 08539 | Error | Error: \$errmsg. |
| 08540 | Error | No port discovered on the device: \$hostaddr. |
| 08541 | Error | No sysobjid found on the device: \$hostaddr. |
| 08542 | Error | Unable to add/refresh the device \$hostaddr: \$errmsg. |
| 08543 | Warning | The route of the device "\$iplnetdev_name" was ignored: a network was already added for another route with the same address and prefix (\$iplnetdevroute_ip_addr/\$iplnetdevroute_prefix). |
| 08544 | Error | Unable to add the subnet-type network for the device "\$iplnetdev_name": the route IP address is set to "default". |
| 08545 | Error | Unable to add the subnet-type network for the device "\$iplnetdev_name": the route prefix is above the limit (\$iplnetdevroute_ip_addr/\$iplnetdevroute_prefix). |
| 08546 | Error | Unable to add the subnet-type network for the device "\$iplnetdev_name": the route refers to non-terminal network (\$iplnetdevroute_ip_addr/\$iplnetdevroute_prefix). |
| 08547 | Error | Unable to add the subnet-type network for the device "\$iplnetdev_name": there is no block able to receive it (\$iplnetdevroute_ip_addr/\$iplnetdevroute_prefix). |
| 08548 | Error | Unable to add the route "\$iplnetdevroute_ip_addr" of the network device "\$iplnetdev_name" as subnet-type network: the route has a prefix "/\$iplnetdevroute_prefix". |
| 08549 | Error | The VLAN ID "\$iplnetdevvlan_number" of the network device "\$iplnetdev_name" was ignored: this ID has already been created in VLAN Manager for another VLAN. |
| 08550 | Error | Unable to add/refresh the device "\$hostaddr": it is a duplicate of the device "\$duplicateof". |
| 08551 | Error | Unable to add the IPAM network(s): no NetChange route matches the selected parameters. |
| 09000 | Notice | Object deleted (\$type). |
| 09002 | Error | Unable to cancel the operation "\$service_name": it cannot be cancelled. |
| 09003 | Notice | The operation "\$service_name" was successfully cancelled. |
| 09004 | Error | Unable to cancel the operation "\$service_name" (\$params): the operation "\$parent_name" (\$parent_parameters) must be cancelled first. |
| 09005 | Error | Unable to cancel the operation "\$service_name": an error occurred (error: \$error, parameters: \$params). |
| 09006 | Error | Unable to cancel the operation "\$service_name" (\$params): it was performed by another user. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 09007 | Error | Unable to delete this object: it is already Free. |
| 09100 | Error | Unable to perform this operation on "\$stat_type" analytics: an error occurred. |
| 10000 | Notice | The custom data was successfully imported. |
| 11001 | Error | The report data is not valid. |
| 11002 | Error | The report file is not valid. |
| 11003 | Error | The report format is not valid. |
| 11004 | Error | The report extension is not valid. |
| 11005 | Error | Can't open report file. |
| 11006 | Error | You must first select reports. |
| 13000 | Error | Unable to perform this operation: you cannot delete the local SOLIDserver. |
| 13001 | Error | Member is not local. |
| 13002 | Error | Member is in another HA group. |
| 13003 | Error | Local member have no IP address configured. |
| 13004 | Error | Too many SOLIDserver appliances configured. HA is designed to set a two-appliances configuration. |
| 13005 | Error | Impossible to switch the remote SOLIDserver to Hot Standby: the local database is in Read-Only. |
| 13006 | Error | Impossible to switch the local SOLIDserver to Hot Standby. |
| 13007 | Error | This SOLIDserver is already one of your remote appliances. |
| 13008 | Error | The remote SOLIDserver is not configured locally yet. |
| 13009 | Error | The IP address used to declare and enroll the remote SOLIDserver does not match the one it is configured with. |
| 13010 | Error | Unable to reset this Master SOLIDserver to Standalone: it is currently associated with a Hot Standby. You need to disable the High Availability configuration before going any further. |
| 13011 | Error | Impossible to restore a backup of this SOLIDserver. It is currently configured in High Availability. You need to break the High Availability setup before going any further. |
| 13012 | Error | Unable to switch the remote SOLIDserver to Hot Standby, the database replication failed. |
| 13013 | Error | Unable to switch the remote SOLIDserver to Hot Standby, the connection was interrupted (Timeout status). |
| 13014 | Error | Unable to find the local SOLIDserver. |
| 13015 | Error | Unable to find the remote SOLIDserver. |
| 13016 | Error | Impossible to add a remote SOLIDserver that is not configured locally. |
| 13017 | Error | Impossible to add a SOLIDserver that is already managed remotely. |
| 13018 | Error | Impossible to add a SOLIDserver that already manages one or more SOLIDserver. |
| 13019 | Error | Impossible to add a remote SOLIDserver to the list: the local SOLIDserver is managed remotely. |
| 13020 | Error | The remote SOLIDserver stopped replicating. |
| 13021 | Error | Unable to add the local SOLIDserver. |
| 13022 | Error | Unable to add the remote SOLIDserver. |
| 13023 | Error | The remote SOLIDserver is unreachable. |
| 13024 | Error | The remote SOLIDserver version is not supported. |
| 13026 | Error | Unable to enroll the remote SOLIDserver: the configuration failed. |
| 13027 | Error | Unable to enroll this SOLIDserver: it is currently being upgraded. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 13028 | Error | Unable to upgrade the remote SOLIDserver: the file containing the local version details cannot be found. |
| 13029 | Error | Unable to enroll the remote SOLIDserver: the local and remote appliances have different architectures (one is a 64-bit and the other 32-bit). |
| 13030 | Notice | Remote SOLIDserver deleted (\$member_name). |
| 13032 | Error | Unable to push the local LDAP/Radius authentication configuration to \$member_name (\$member_hostaddr): this remote SOLIDserver is upgrading. Try again once the upgrade is over. |
| 13033 | Notice | The local LDAP/Radius authentication configuration was successfully pushed to the remote SOLIDserver \$member_name (\$member_hostaddr). |
| 13034 | Error | Unable to enroll the remote appliance as Hot Standby: the enrollment is locked. |
| 13035 | Error | Unable to switch the local SOLIDserver to Master: \$errmsg. |
| 13036 | Error | Unable to launch the upgrade: the Hot Standby is unreachable. |
| 13037 | Error | Unable to upgrade the Hot Standby: the maintenance period has expired. |
| 13038 | Error | Unable to upgrade: the maintenance period has expired. |
| 13039 | Error | Unable to upgrade the Hot Standby: an error occurred. |
| 13040 | Error | Unable to upgrade the Hot Standby: its license is not valid or missing. |
| 13041 | Error | Unable to upgrade: the license is not valid or missing. |
| 13042 | Error | Unable to upgrade the remote SOLIDserver "\$name" (\$ipaddr): its license is not valid or missing. |
| 13043 | Error | Unable to upgrade the remote SOLIDserver "\$name" (\$ipaddr): its maintenance period has expired. |
| 13045 | Error | Unable to switch the local SOLIDserver to master: \$member is being enrolled as Hot Standby. |
| 13046 | Error | Unable to switch the local SOLIDserver to master: \$member is already switching. |
| 13047 | Error | Unable to enroll the remote SOLIDserver as Hot Standby: \$member is already enrolled. |
| 13048 | Error | Unable to enroll the remote appliance as Hot Standby: the database encryption cannot be activated on this appliance. |
| 13049 | Error | Unable to enroll this SOLIDserver as Hot Standby: the database encryption key is missing. |
| 13050 | Error | Unable to enroll this SOLIDserver as Hot Standby: the database encryption key file is corrupted, please try again. |
| 13051 | Error | Unable to switch the local SOLIDserver to Standalone: "\$member" is already switching its role. |
| 13052 | Error | Unable to perform the operation: the SOLIDserver "\$member" is currently being upgraded. |
| 13054 | Error | Unable to perform the operation "\$operation" on the SOLIDserver "\$member_name": it can only be performed on "\$member_type" SOLIDservers set with one of the following statuses \$member_states. |
| 13055 | Error | Unable to upgrade "\$name": you cannot upgrade the local SOLIDserver using the option "Upgrade remote SOLIDserver". |
| 13057 | Error | Unable to perform this operation on SOLIDserver "\$member_name": its IP address is invalid. |
| 13058 | Error | Unable to perform the operation "\$operation" on the SOLIDserver "\$member_name": you cannot perform it on "\$member_type" SOLIDservers. |
| 13059 | Error | Unable to perform the operation "\$operation" on the SOLIDserver "\$member_name": it is not defined in MemberLock.php. |
| 13060 | Error | Unable to upgrade: there is not enough disk space. The operation was cancelled. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 13061 | Notice | The security events collection was successfully enabled. |
| 13062 | Notice | The security events collection was successfully disabled. |
| 13063 | Error | Unable to perform this operation: the version of the selected SOLIDserver does not support security events collection. |
| 13064 | Error | Unable to upgrade: some checks failed on the Hot Standby. The upgrade was cancelled. |
| 13065 | Error | Unable to delete this SOLIDserver: "\$member_name" is busy. |
| 13066 | Error | The remote SOLIDserver does not support IPv6 management. |
| 13067 | Error | Unable to edit this SOLIDserver: the remote serial number (\$new_member_serial) does not match the expected one (\$member_serial). |
| 14401 | Error | Invalid user/password. |
| 14404 | Error | Command not supported on this server. |
| 20001 | Error | Cannot find master zone (\$dnszone_name). |
| 20002 | Error | Cannot choose the master zone. |
| 20003 | Error | Invalid RR. |
| 20004 | Error | The server didn't answer in time to the request for a new RR. |
| 20005 | Error | The server didn't answer in time to the RR deletion request. |
| 20006 | Error | There is more than one zone available for delegation. |
| 20007 | Error | The server did not answer in time to the request for a new zone. |
| 20008 | Error | The server did not answer in time to the zone deletion request. |
| 20009 | Error | The server did not answer in time to the request for a new ACL. |
| 20010 | Error | The server did not answer in time to the ACL deletion request. |
| 20011 | Error | Cannot delete the last NS RR. |
| 20012 | Error | DNS zone already exists. |
| 20016 | Error | Unable to add the resource record "\$rr_name": it cannot belong to the DNS zone "\$dnszone_name". |
| 20017 | Error | The specified DNS server, zone or record does not exist. |
| 20020 | Error | You must select DNS zones. |
| 20021 | Error | You must select DNS records. |
| 20022 | Error | Cannot migrate a SOA or NS RR. |
| 20023 | Error | You must select a DNS master zone. |
| 20024 | Error | The zone does not exist. |
| 20025 | Notice | DNS RR deleted (name: \$value1, zone: \$dnszone_name, view: \$dnsview_name, server: \$dns_name). |
| 20026 | Notice | DNS zone deleted (name: \$dnszone_name, view: \$dnsview_name, server: \$dns_name). |
| 20027 | Notice | DNS view deleted (name: \$dnsview_name server: \$dns_name). |
| 20028 | Notice | The DNS server "\$dns_name" was successfully deleted. |
| 20029 | Notice | DNS zone enabled (name: \$dnszone_name, view: \$dnsview_name, server: \$dns_name). |
| 20030 | Notice | DNS zone disabled (name: \$dnszone_name, view: \$dnsview_name, server: \$dns_name). |
| 20031 | Notice | DNS zone synchronized (name: \$dnszone_name view: \$dnsview_name server: \$dns_name). |
| 20032 | Notice | The DNS server "\$dns_name" was successfully synchronized. |
| 20033 | Error | There is no view for this DNS server. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 20034 | Notice | The record "\$rr_name" (\$rr_type) was successfully added in the zone "\$dnszone_name" of the server "\$dns_name". |
| 20035 | Notice | The zone "\$dnszone_name" (\$dnszone_type) was successfully added on the server "\$dns_name". |
| 20036 | Error | You must select a slave DNS zone. |
| 20037 | Error | This server is not an EfficientIP DNS server. |
| 20039 | Notice | DNS zone added. |
| 20040 | Error | You must select a slave/stub DNS zone. |
| 20041 | Error | The DNS zone is empty. |
| 20042 | Notice | DNS refresh. |
| 20043 | Notice | DNS notify. |
| 20044 | Notice | DNS retransfer. |
| 20045 | Error | The value to Replace was not found. |
| 20046 | Notice | DNS cache flushed. |
| 20047 | Notice | The DNS querylog command has been toggled. |
| 20048 | Notice | DNS view "\$dnsview_name" added. |
| 20049 | Notice | DNS key "\$key_name" added. |
| 20051 | Error | The target zone \$dnszone_name does not exist and you have not granted external domain management. |
| 20052 | Notice | The DNS ACL "\$dnsacl_name" was successfully added. |
| 20053 | Error | The RR A \$rr_name in the zone \$dnszone_name does not exist. |
| 20054 | Notice | DNS view updated. |
| 20055 | Error | DNS key is used (\$object_type \$object_name). |
| 20056 | Notice | DNS ACL deleted. |
| 20057 | Notice | DNS KEY deleted. |
| 20058 | Error | DNS ACL is used (\$object_type \$object_name). |
| 20060 | Error | DNS name already exist. |
| 20061 | Error | RR name already used (name:\$rr_glue zone:\$dnszone_name type:\$rr_type). |
| 20062 | Error | Cannot delete a child DNS server. |
| 20063 | Error | Cannot delete a SMART DNS server with children. |
| 20064 | Error | Cannot change the RR type. |
| 20065 | Error | DNS server name already exist (name: \$dns_name). |
| 20066 | Error | DNS server address already exists (address: \$hostaddr). |
| 20067 | Error | Cannot change DNS server type (from: \$from, to: \$to). |
| 20068 | Notice | The DNS server "\$dns_name" was successfully added. |
| 20069 | Error | DNS server is in read-only mode (SMART DNS member). |
| 20071 | Error | The filter validation not allowed this RR "\$rr_name". |
| 20072 | Error | You do not have the right to handle this IP address "\$ip_address". |
| 20073 | Error | You do not have the right to handle this IP address pointed by this RR CNAME or the RR A does not exist. |
| 20074 | Error | The filter ban not allowed this RR "\$rr_name". |
| 20076 | Error | Could not find the specified DNS server. |
| 20077 | Error | Could not find the specified DNS zone. |
| 20078 | Error | Cannot delete the ACL "\$dnsacl_name". |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 20079 | Error | Cannot delete the parameter "\$param_key", it doesn't exist. |
| 20080 | Error | The specified parameter cannot be deleted as it doesn't exist. |
| 20081 | Error | Cannot delete SOA. |
| 20082 | Error | Unable to perform the command: this DNS control command is not supported by the server. |
| 20084 | Warning | Existing A RR with same hostname (RR:\$rr_value1). |
| 20085 | Warning | Existing A RR with same IP address (shortname: \$rr_glue , zone name: \$dnszone_name , server: \$dns_name). |
| 20086 | Error | Views are not supported on this kind of DNS server. |
| 20087 | Error | Invalid property name. |
| 20088 | Warning | The option "\$option" is not supported for the object "\$objectname" (\$type). This option will be removed from the database. |
| 20089 | Error | The ACL "\$name" is not properly configured. |
| 20090 | Notice | The DNS zone was successfully added. |
| 20091 | Notice | The DNS view was successfully added. |
| 20092 | Error | The DNS RR \$rr_name already exists. |
| 20093 | Warning | The DNS option \$option is not supported. |
| 20094 | Warning | The DNS option \$option is not supported on the view \$dnsview_name. |
| 20095 | Warning | The DNS option \$option is not supported on the \$dnszone_type zone \$dnszone_name. |
| 20096 | Error | Unable to add the record "\$rr_name": this name is already used for a CNAME/DNAME record. |
| 20097 | Error | Unable to move a DNS server from one smart to the other: you cannot move the server "\$dns_id" from the smart "\$old_vdns_parent_id" to the smart "\$new_vdns_parent_id". |
| 20098 | Error | Unable to replace the A RR value with "\$value1": the new value must be a valid IP address. |
| 20099 | Error | Unable to replace the A RR value: the new value must be a valid IPv6 address. |
| 20100 | Error | Unable to edit the ACL "\$acl_name": you cannot edit a predefined ACL. |
| 20101 | Error | Unable to add the zone: "\$zone_type" zones are not supported on "\$dns_type" servers. |
| 20111 | Error | Unable to add the zone "\$dnszone_name": you cannot manage more than 32 RPZ zones in one view or server (with no view). |
| 20112 | Error | Unable to add the record: the parameter \$parameters (\$param_value) is not supported on \$rr_type records. |
| 20113 | Error | Unable to perform this operation: you cannot change a server "\$from_type" to "\$to_type". |
| 20114 | Error | Unable to add the "\$rr_type" record (value: "\$rr_value") in the zone "\$zone_name": you cannot add more than "\$rr_limit" records in one RRset. |
| 20116 | Error | Unable to convert the DNS server "\$dns_name" into a smart. |
| 20117 | Error | This resource record does not exist. |
| 20118 | Error | Unable to convert the zone "\$dnszone_name": the SOA record is missing. |
| 20120 | Error | You cannot delete the keytab "\$object_name": it is used by the DNS server "\$dns_name". |
| 20121 | Error | Unable to disable the GSS-TSIG key: it is still used by a zone. |
| 20122 | Error | This DNS server does not support GSS-TSIG. |
| 20123 | Error | This zone does not support GSS-TSIG. |

| Code | Level | Description |
|-------------|--------------|--|
| 20124 | Error | Unable to edit the update-policy: no keytab is selected on the server "\$dns_name". |
| 20125 | Error | Unable to edit the update-policy of the zone "\$dnszone_name": GSS-TSIG is not enabled on the server "\$dns_name". |
| 20126 | Error | Unknown DNS view. |
| 20130 | Error | The engine of the DNS server "\$dns_name" does not support RRL. |
| 20132 | Error | Unable to convert to master the zone "\$dnszone_name" in the smart architecture "\$dns_name": no matching zone was found in the physical server(s). |
| 20133 | Error | Unable to perform the command: smart architectures do not support DNS control commands. |
| 20134 | Error | Unable to edit the server "\$dns_name", it is in read-only: Microsoft DNS servers with agent are no longer supported. |
| 20135 | Error | Unable to convert the smart architecture: it contains too many servers (\$count instead of \$expected). |
| 20136 | Error | Unable to set DDNS scavenging: the DNS zone "\$dnszone_name" (server: \$dns_name) does not support it. |
| 20137 | Error | Unable to edit the DDNS scavenging of the DNS zone "\$dnszone_name" (server: \$dns_name). |
| 20138 | Notice | The DDNS scavenging was successfully edited for the DNS zone "\$dnszone_name" (server: \$dns_name). |
| 20189 | Error | Unable to import the archive: the ACL "\$dnsacl_name" already exists. |
| 20190 | Notice | The option "\$param_key" was successfully added with the value "\$param_value". |
| 20191 | Error | Unable to add the view "\$dnsview_name" on the server "\$dns_name": views are not supported on "\$dns_type" cloud servers. |
| 20192 | Error | Unable to add/edit the record "\$rr_name" on the server "\$dns_name": you cannot add or edit NS records on cloud servers. |
| 20193 | Error | Unable to add this Amazon Route 53 server: "\$dns_name" already exists. |
| 20194 | Error | Unable to add the record "\$rr_name": "\$rr_type" records are not compatible with "\$dns_type" Cloud servers. |
| 20195 | Error | Unable to add the zone "\$dnszone_name": "\$dnszone_type" zones are not compatible with "\$dns_type" Cloud servers. |
| 20196 | Notice | The NS record "\$ns_value" was successfully replicated on the smart server. |
| 20197 | Error | Unable to replicate the NS records: "\$dns_name" is not a Cloud server. |
| 20198 | Error | Unable to replicate the NS records: "\$dns_name" is not a smart server. |
| 20199 | Error | Unable to delete the records from the server "\$dns_name": you cannot delete NS records from a Cloud server. |
| 20200 | Error | Unable to add/edit the DNS server "\$dns_name": the SNMP parameters cannot be saved. |
| 20201 | Error | The DNS ACL does not exist. |
| 20202 | Error | Unable to remove "admin" from the list: this ACL is mandatory for allow-update. |
| 20203 | Notice | DNS server modified (name: \$dns_name). |
| 20204 | Error | The synchronization of the DNS server "\$dns_name" failed: connection timeout. |
| 20205 | Error | The synchronization of the DNS server "\$dns_name" failed: invalid credentials. |
| 20206 | Warning | The DNS server is already managed by an appliance ("\$serial" at "\$hostaddr") and was last synchronized "\$time" second(s) ago. Click on OK to force management from the new appliance. |
| 20207 | Error | Unable to add/edit the DNS server ("\$hostaddr"): it is unreachable. |
| 20208 | Error | Unable to add the DNS server ("\$hostaddr"): it does not match the selected type (EIP DNS / EIP DNS Package). |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 20209 | Error | Unable to add the DNS server ("\$hostaddr"): the security enrollment failed. |
| 20210 | Warning | Unable to merge the zones content: the source zone "\$dnszone_name" is empty. |
| 20211 | Warning | The record "\$rr_name" (type: \$rr_type, value: \$rr_value) will be deleted from the zone "\$dnszone_name" (server: \$dns_name, view: \$dnsview_name). |
| 20212 | Warning | The record "\$rr_name" (type: \$rr_type, value: \$rr_value) will be added to the zone "\$dnszone_name" (server: \$dns_name, view: \$dnsview_name). |
| 20213 | Error | Unable to add the Azure DNS server "\$dns_name": the specified Azure account is already used. |
| 20214 | Error | Unable to add the Azure DNS server: the specified Azure resource group is already used by the server "\$dns_name". |
| 20215 | Notice | The option "\$option_name" was successfully edited (new value: \$option_value). |
| 20300 | Warning | You can add the view "\$dnsview_name" to the smart architecture "\$dns_name", but views are not supported by at least one of its physical servers. Do you really want to proceed?. |
| 20301 | Warning | You can add the zone "\$dnszone_name" to the smart architecture "\$dns_name", but this zone type (\$dnszone_type) is not supported by at least one of its physical servers. Do you really want to proceed?. |
| 20302 | Warning | You can add the RPZ zone "\$dnszone_name" (type: "\$dnszone_type") to the smart architecture "\$dns_name", but RPZ zones are not supported by at least one of its physical servers. Do you really want to proceed?. |
| 20303 | Warning | You can add the record "\$rr_name" to the smart architecture "\$dns_name", but this record type (\$rr_type) is not supported by at least one of its physical servers. Do you really want to proceed?. |
| 20304 | Error | Unable to add the view "\$dnsview_name" on the smart architecture "\$dns_name": views are not supported by at least one of its physical servers. |
| 20305 | Error | Unable to add the zone "\$dnszone_name" (type: \$dnszone_type) on the smart architecture "\$dns_name": this zone type is not supported by at least one of its physical servers. |
| 20306 | Error | Unable to add the RPZ zone "\$dnszone_name" (type: "\$dnszone_type") on the smart architecture "\$dns_name": RPZ zones are not supported by at least one of its physical servers. |
| 20307 | Error | Unable to add the record "\$rr_name" (type: \$rr_type) on the smart architecture "\$dns_name": this record type is not supported by at least one of its physical servers. |
| 20308 | Error | Unable to add the view "\$dnsview_name" on the server "\$dns_name": views are not supported by "\$dns_type" servers. |
| 20309 | Error | Unable to add the zone "\$dnszone_name" (type: "\$dnszone_type") on the server "\$dns_name": this zone type is not supported by "\$dns_type" servers. |
| 20310 | Error | Unable to add the RPZ zone "\$dnszone_name" (type: "\$dnszone_type") on the server "\$dns_name": RPZ zones are not supported by "\$dns_type" servers. |
| 20311 | Error | Unable to add the record "\$rr_name" (type: "\$rr_type") on the server "\$dns_name": this record type is not supported by "\$dns_type" servers. |
| 20313 | Warning | The view "\$dnsview_name" is not supported by the server "\$dns_name". |
| 20314 | Warning | The zone "\$dnszone_name" (type: "\$dnszone_type") is not supported by the server "\$dns_name". |
| 20315 | Warning | The RPZ zone "\$dnszone_name" (type: "\$dnszone_type") is not supported by the server "\$dns_name". |
| 20316 | Warning | The record "\$rr_name" (type: "\$rr_type") in the zone "\$dnszone_name" is not supported by the server "\$dns_name". |
| 20317 | Error | Unable to generate the report: you must select a smart architecture. |
| 20318 | Error | Unable to add/edit the record: you cannot add or edit records in "\$dnszone_type" zones. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 20319 | Error | Unable to delete the record: you cannot delete records from "\$dnszone_type" zones. |
| 20320 | Error | Unable to add the DNS server into the architecture: the server name does not exist. |
| 20321 | Error | Unable to add the DNS server into the architecture: the server is managed by another architecture or is an architecture. |
| 20322 | Error | Unable to add/edit the server: the server "\$dns_name" already exists, it is configured with a different delegation set. |
| 20323 | Error | Unable to add/edit the server: the public server "\$dns_name" already exists, you cannot convert it into a private server. |
| 20324 | Error | Unable to add the server: the specified VPC list is already used on another AWS private server on this account. |
| 20325 | Error | Unable to add the server: the specified virtual network (VNet) list is already used on another Azure private server of this account. |
| 20327 | Error | Unable to add the server: the specified reusable delegation set ID (\$delegation_set) is already configured on another public AWS server of this account. |
| 20328 | Error | Unable to add the server: another public AWS server configured without reusable delegation set already exists on this account. |
| 20329 | Error | Unable to add/edit the server: the number of selected virtual networks exceeds the authorized limit. |
| 20330 | Error | Unable to add/edit the server: the number of selected VPCs exceeds the authorized limit. |
| 20331 | Error | Unable to add/edit the server: the private server "\$dns_name" already exists, you cannot convert it into a public server. |
| 20332 | Error | Unable to delete the DNS zone "\$dnszone_name": it is associated with the Guardian list "\$glist_name". |
| 20333 | Warning | Incompatibility found: the zone "\$dnszone_name" is not a TLD, which is not supported by the server "\$dns_name". |
| 20334 | Error | Unable to perform this command: the remote server only supports BIND control commands. |
| 21100 | Error | Unable to find the Guardian policy. |
| 21101 | Error | Unable to delete the Guardian policy: it is in read-only. |
| 21102 | Error | Unable to edit the Guardian policy: it is in read-only. |
| 21103 | Error | Unable to delete the Guardian policy: you cannot delete a policy instance. |
| 21104 | Error | Unable to delete the Guardian policy: it is associated with one or more servers. |
| 21105 | Error | Unable to duplicate the Guardian policy: you cannot duplicate a policy instance. |
| 21106 | Error | This Guardian policy already exists. |
| 21107 | Error | The server is already associated with a Guardian policy. |
| 21108 | Error | Unable to associate this policy with the selected server: it does not exist, does not support the policy or is not in your resources. |
| 21109 | Error | Unable to edit the Guardian policy: deployed policies are read-only. |
| 21110 | Notice | The Guardian policy was successfully deleted. |
| 21111 | Notice | The Guardian policy was successfully created. |
| 21201 | Error | Unable to delete the trigger: it belongs to a read-only Guardian policy. |
| 21202 | Error | Unable to find the Guardian policy. |
| 21203 | Error | Unable to find the Guardian trigger. |
| 21204 | Error | Unable to delete the Guardian trigger: it belongs to a deployed policy in read-only. |
| 21205 | Error | This Guardian trigger already exists. |
| 21206 | Error | This position was already assigned to the Guardian trigger "\$name". |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 21207 | Error | This position can only be assigned to a default Guardian trigger. |
| 21208 | Error | Unable to add or edit the Guardian trigger in a deployed policy. |
| 21209 | Error | The rule syntax "\$rule" is invalid: \$syntax_error_msg. |
| 21210 | Notice | The Guardian trigger was successfully deleted. |
| 21211 | Notice | The Guardian trigger was successfully created. |
| 21212 | Notice | The Guardian trigger is now Managed. |
| 21213 | Notice | The Guardian trigger is now Unmanaged. |
| 21303 | Notice | Guardian clients flushed. |
| 21304 | Notice | Guardian cache flushed. |
| 21305 | Error | Unable to edit the Guardian parameter "\$param_key": its value (\$param_value) has an invalid format. |
| 21306 | Error | Unable to send the command: DNS Guardian is not running. |
| 21307 | Notice | The Guardian cache entries were successfully set as expired. |
| 21500 | Notice | The ruleset "\$guardianruleset_name" was successfully added. |
| 21501 | Notice | The ruleset "\$guardianruleset_name" was successfully deleted. |
| 21502 | Notice | The ruleset was successfully imported. |
| 21503 | Error | Unable to find the ruleset. |
| 21504 | Error | Unable to add the ruleset "\$guardianruleset_name": it already exists. |
| 21505 | Error | Unable to edit the ruleset: deployed rulesets are read-only. |
| 21506 | Error | Unable to delete the ruleset: you cannot delete a ruleset instance. |
| 21507 | Error | Unable to associate the ruleset with the selected server: it does not exist, does not support rulesets or is not in your resources. |
| 21508 | Error | Unable to delete the ruleset: it is associated with one or more servers. |
| 21509 | Error | Unable to add the ruleset: another ruleset is already associated with the server and view "\$param_value". |
| 21510 | Warning | The ruleset "\$guardianruleset_name" was added, but the view "\$guardianruleset_view_name" does not exist on at least one of the selected Guardian servers (\$dns_name_list). |
| 21511 | Warning | The ruleset "\$guardianruleset_name" was added, but the view "\$guardianruleset_view_name" is not one of the 16 first views on at least one of the Selected Guardian servers (\$dns_name_list). |
| 21512 | Error | Unable to duplicate the ruleset: you cannot duplicate a ruleset instance. |
| 21513 | Error | Unable to add the rule "\$guardianrule_name": it already exists. |
| 21514 | Error | Unable to find the rule. |
| 21515 | Notice | The rule "\$guardianrule_name" was successfully added. |
| 21516 | Notice | The rule "\$guardianrule_name" was successfully deleted. |
| 21517 | Notice | Rule enabled. |
| 21518 | Notice | Rule disabled. |
| 21519 | Notice | The rule was successfully imported. |
| 21520 | Error | Unable to add the rule, its definition is incorrect: the list "\$glist_name_list" does not exist. |
| 21521 | Error | Unable to add the rule, its definition is incorrect: the tag "\$tag_name_list" does not exist. |
| 21522 | Error | Unable to add the list "\$guardianglist_name": it already exists. |
| 21523 | Error | Unable to find the list. |
| 21524 | Notice | The list "\$guardianglist_name" was successfully added. |

| Code | Level | Description |
|-------------|--------------|---|
| 21525 | Notice | The list "\$guardianglist_name" was successfully deleted. |
| 21526 | Error | Unable to add the list: another list is already associated with the zone "\$guardianglist_zone_name". |
| 21527 | Error | Unable to rename the list "\$guardianglist_name": it is used in at least one rule or trigger. |
| 21528 | Error | Unable to add the list: you reached the maximum number of lists (\$max_limit). |
| 21529 | Error | Unable to delete the list "\$guardianglist_name": it is used in at least one rule or trigger. |
| 21530 | Error | Unable to delete the list: you cannot delete a list instance. |
| 21531 | Notice | The list entry "\$guardianglistentry_identifier1 \$guardianglistentry_identifier2 \$guardianglistentry_identifier3" was successfully deleted. |
| 21532 | Notice | The list entry "\$guardianglistentry_value1 \$guardianglistentry_value2 \$guardianglistentry_value3" was successfully added. |
| 21533 | Error | Unable to find the list entry. |
| 21534 | Error | Unable to edit the list entry "\$guardianglistentry_value": you cannot set an entry with the status Bad format. |
| 21535 | Error | Unable to edit the list entry "\$guardianglistentry_value": it matches a CNAME record in the DNS. |
| 21536 | Error | Unable to add the list entry: the identifier(s) of an entry cannot all be *. |
| 21537 | Error | Unable to add the list entry: several identifiers can contain * if they follow each other. |
| 21538 | Error | Unable to add the list entry: an entry cannot be added in an External list or point to a Slave zone. |
| 21539 | Notice | The tag "\$guardiantag_name" was successfully added. |
| 21540 | Notice | The tag "\$guardiantag_name" was successfully deleted. |
| 21541 | Error | Unable to delete the tag "\$guardiantag_name": it is used in at least one rule. |
| 21542 | Notice | The tag was successfully imported. |
| 21543 | Error | Unable to add the tag: you cannot manage more than "\$max_limit" tags. |
| 21544 | Error | Unable to add the tag "\$guardiantag_name": it already exists. |
| 21545 | Error | Unable to find the tag. |
| 21546 | Error | Unable to manage the rule "\$guardianrule_name": you cannot manage more than "\$max_limit" rules per ruleset. |
| 21547 | Error | Unable to manage the rule "\$guardianrule_name": the view "\$view_name" of the action Jump to view does not exist on the Guardian server "\$dns_name" of the ruleset. |
| 21548 | Error | Unable to manage the rule "\$guardianrule_name": the view "\$view_name" of the action Jump to view is not one of the 16 first views of the Guardian server "\$dns_name" of the ruleset. |
| 21549 | Error | Unable to edit the ruleset "\$guardianruleset_name": the view(s) "\$views_name" set in the action Jump to view of at least one rule do not exist on the Guardian server "\$dns_name". |
| 21550 | Error | Unable to edit the ruleset "\$guardianruleset_name": the view(s) "\$views_name" set in the action Jump to view of at least one rule are not part of the 16 first views of the Guardian server "\$dns_name". |
| 21551 | Error | Unable to add the trigger, its definition is incorrect: the list "\$glist_name_list" does not exist. |
| 21552 | Error | Unable to add the list "\$listName": the position "\$listPos" is already used. |
| 22100 | Error | Can't read file '\$file'. |
| 22101 | Error | Archive file is unreadable, and may be corrupted. |
| 22102 | Error | Can't find file '\$file'. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 22103 | Error | Invalid domain name: \$value. |
| 22104 | Error | Invalid TTL value (\$param \$value). |
| 22105 | Error | Invalid RR type (\$param \$value). |
| 22106 | Error | Unsupported file format (\$extension). |
| 22107 | Error | Missing value (\$param). |
| 22109 | Error | No master server specified for zone \$name. |
| 22110 | Error | Invalid value (\$param \$value). |
| 22111 | Error | Invalid address. |
| 22112 | Error | IP address is outside a network or pool. |
| 22113 | Error | Size must be greater than zero. |
| 22114 | Error | Missing address. |
| 22115 | Error | Range is outside scope. |
| 22116 | Error | Pool is outside a network. |
| 22117 | Error | Network is outside a parent network. |
| 22118 | Error | Invalid MAC address (\$value). |
| 22119 | Error | Size is too long (\$param: \$value). |
| 22120 | Error | The RR is not in the current zone. |
| 22121 | Error | Incorrect values: "\$dump". |
| 22122 | Error | Invalid gateway. |
| 22200 | Error | The column is not present in the CSV file. |
| 22201 | Error | A mandatory parameter (\$param) is missing. |
| 22202 | Error | Invalid DHCP option code (\$code). |
| 22203 | Error | Invalid client identifier \$client_ident). |
| 22204 | Error | No DHCP static identifier. |
| 22205 | Error | Bad name format for static "\$static_name". |
| 22206 | Error | Unable to activate the license on the appliance "\$name" (\$ipaddr): a license is already installed. |
| 22207 | Error | Unable to perform the operation on the appliance "\$name" (\$ipaddr): no license was found. |
| 22208 | Error | Unable to activate the license on the appliance "\$name" (\$ipaddr): the appliance is unreachable. |
| 22209 | Error | Unable to activate the license on the appliance "\$name" (\$ipaddr): the license is invalid. |
| 22210 | Error | Unable to activate the license on the appliance "\$name" (\$ipaddr): this operation is not supported. |
| 22211 | Error | Unable to activate the license on the appliance "\$name" (\$ipaddr): the license has expired. |
| 22212 | Error | Unable to activate the license on the appliance "\$name" (\$ipaddr): the license has not started yet. |
| 22213 | Error | Unable to import the file: the class parameter "\$parameter" is missing its inheritance and/or propagation property. |
| 22214 | Error | Unable to import the file: the mandatory column "\$param" is missing. |
| 35000 | Notice | The IPv4 address was successfully imported. |
| 35001 | Notice | The IPv6 address was successfully imported. |
| 35002 | Notice | The IPv4 network "\$block_name" was successfully imported. |
| 35003 | Notice | The IPv6 network was successfully imported. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 35004 | Notice | The range was successfully imported. |
| 35005 | Notice | The group was successfully imported. |
| 35006 | Notice | The IPv4 pool was successfully imported. |
| 35007 | Notice | The IPv6 pool was successfully imported. |
| 35008 | Notice | The RR was successfully imported. |
| 35009 | Notice | The scope was successfully imported. |
| 35010 | Notice | The space was successfully imported. |
| 35011 | Notice | The static was successfully imported. |
| 35012 | Notice | The IPv4 network was successfully imported. |
| 35013 | Notice | The IPv6 network was successfully imported. |
| 35014 | Notice | The user was successfully imported. |
| 35015 | Notice | The zone was successfully imported. |
| 35016 | Notice | The IPv4 address and alias were successfully imported. |
| 35017 | Notice | The RR "\$rr_name" was successfully checked. |
| 35018 | Notice | The zone "\$dnszone_name" was successfully checked. |
| 35019 | Notice | Line is correct. |
| 35020 | Notice | The custom data was successfully imported. |
| 35021 | Notice | Finished importing MS DHCP configuration. |
| 35022 | Notice | The VRF "\$vrftobject_name" was successfully imported. |
| 35023 | Notice | The domain "\$vlmdomain_name" was successfully imported. |
| 35024 | Notice | The range "\$vlmrange_name" was successfully imported. |
| 35025 | Notice | The vlan "\$vlmvlan_vlan_id" was successfully imported. |
| 35026 | Notice | The VRF Route Target "\$src_rd_id"/"\$dest_rd_id" was successfully imported. |
| 35027 | Notice | The license was successfully activated on the appliance "\$name" (\$ipaddr). |
| 35028 | Notice | The IPv4 alias was successfully imported. |
| 35029 | Notice | The IPv6 alias was successfully imported. |
| 35100 | Warning | Ignoring all SOA. |
| 35101 | Warning | Zone [\$name] empty (no file associated). |
| 35102 | Warning | The record "\$name" was not imported: its name does not comply with RFC 1034. |
| 35103 | Warning | Root zones are not handled. |
| 35104 | Warning | This static name is already used: the new static will be named "\$dhcphost_newname" instead of "\$dhcphost_oldname". |
| 35200 | Warning | Multiple sizes have been specified in CSV file. |
| 36000 | Error | The package "\$package" version "\$version" was successfully added. |
| 36004 | Notice | The package "\$package" version "\$version" was successfully deleted. |
| 36005 | Notice | The package "\$package" version "\$version" was successfully installed. |
| 36006 | Error | Unable to install the package: it does not exist. |
| 36007 | Error | Unable to delete the package "\$package" version "\$version": it is already installed. |
| 36008 | Notice | The package "\$package" version "\$version" was successfully uninstalled. |
| 36009 | Notice | The package "\$archive" was successfully added. |
| 36010 | Error | Unable to install the package "\$package" version "\$version": it is already installed. |
| 36011 | Error | Unable to uninstall the package "\$package" version "\$version": it is already uninstalled. |
| 36014 | Error | Unable to add the package "\$package" version "\$version": it already exists. |

| Code | Level | Description |
|-------------|--------------|--|
| 36016 | Error | Unable to delete the package "\$package" in version "\$version": you cannot delete an already installed package. Uninstall the package before deleting it. |
| 36017 | Error | Unable to install the package "\$package" in version "\$version": you cannot install a package containing system files such as "\$path". |
| 36018 | Error | Package processing error. |
| 36019 | Error | Unable to add the package "\$archive": its format is invalid. |
| 36020 | Error | Unable to add the package: no manifest found. |
| 36021 | Error | Unable to add the package: the manifest is not a valid JSON file. |
| 36022 | Error | Unable to add the package: the manifest is missing a mandatory key. |
| 36023 | Error | Unable to add the package: the manifest version is not supported. |
| 36024 | Error | Unable to add the package: the value of the manifest key "\$key" is invalid (\$value). |
| 36025 | Error | Unable to add the package: the format of the dependency "\$dependency" is invalid (\$description). |
| 36026 | Error | Unable to add the package: the manifest mentions a file "\$file" not present in the archive. |
| 36027 | Error | Unable to add the package: it contains the file "\$file" that cannot be installed in this location. |
| 36028 | Error | Unable to add the package: the manifest cannot be encoded for storage. |
| 36029 | Error | Unable to add the package: the file hash of "\$file" does not match. |
| 36030 | Error | Unable to add the package: it contains a file "\$file" not listed in the manifest. |
| 36031 | Error | Unable to add the package: it cannot be stored. |
| 36032 | Error | Unable to add the package: invalid file entry. |
| 36033 | Error | Unable to install the package: the dependency "\$package" is missing and must satisfy the constraint "\$constraint". |
| 36034 | Error | Unable to install the package: the dependency "\$package" does not satisfy the constraint "\$constraint". |
| 36035 | Error | Unable to perform the operation on the package: it is already ongoing. |
| 36036 | Error | Unable to install the package: the install script cannot be extracted. |
| 36037 | Error | Unable to install the package: its content cannot be extracted. |
| 36038 | Error | Unable to install the package: the pre-installation checks failed. |
| 36039 | Error | Unable to install the file "\$file": it already exists. |
| 36040 | Error | The capability "\$supported_element" is required. |
| 36041 | Error | The capability "\$supported_element" does not satisfy the constraint "\$predicate". |
| 36042 | Error | Unable to uninstall the package: it is required by "\$package". |
| 36043 | Warning | The package was installed, but the post-installation script returned an error. |
| 36044 | Warning | The package was uninstalled, but the post-uninstallation script returned an error. |
| 36045 | Error | Unable to uninstall the package: the pre-uninstallation script failed. |
| 37000 | Notice | The RR was successfully copied. |
| 37001 | Notice | The RR was successfully migrated. |
| 38000 | Notice | The IP address was successfully migrated. |
| 38001 | Notice | The network was successfully migrated. |
| 38002 | Notice | The network was successfully split. |
| 38003 | Notice | The network was successfully migrated. |
| 38004 | Notice | The network was successfully migrated. |
| 38005 | Notice | The pool was successfully migrated. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 38006 | Notice | The IP address was successfully migrated. |
| 38007 | Notice | The old network was successfully deleted. |
| 38008 | Notice | The old network was successfully deleted. |
| 38009 | Notice | The old pool was successfully deleted. |
| 38010 | Notice | The old IP address was successfully deleted. |
| 38011 | Notice | The IP address was successfully migrated. |
| 38012 | Notice | The old IP address was successfully deleted. |
| 38013 | Notice | The space "\$site_name" was successfully deleted. |
| 38014 | Notice | The network "\$block_name" was successfully deleted. |
| 38015 | Notice | The network "\$subnet_name" was successfully deleted. |
| 38016 | Notice | The pool "\$pool_name" was successfully deleted. |
| 38017 | Notice | The IPv4 address was successfully deleted (space: \$site_name, parent network: \$block_name, network: \$subnet_name, address: \$ip_addr, name: \$name). |
| 38018 | Notice | The network \$block_name is already a VLSM. |
| 38019 | Notice | The VLSM network "\$block_name" was successfully added. |
| 38020 | Notice | The network "\$block_name" was successfully added. |
| 38021 | Notice | The network "\$subnet_name" was successfully added. |
| 38022 | Notice | The IPv4 pool "\$pool_name" was successfully added. |
| 38023 | Notice | The IPv4 address "\$ip_name" was successfully added. |
| 38024 | Notice | The alias "\$ip_name" was successfully added. |
| 38025 | Notice | The name of the IP address was successfully edited: "\$ip_name". |
| 38026 | Notice | No change required on that IP address. |
| 38027 | Notice | The IPv6 address was successfully deleted (space: \$site_name, parent network: \$block6_name, network: \$subnet6_name, address: \$ip6_addr, name: \$ip6_name). |
| 38028 | Notice | The space "\$site_name" was successfully added. |
| 38029 | Notice | The IPv6 address "\$ip6_name" was successfully added. |
| 38030 | Notice | The network \$subnet6_name has been successfully added. |
| 38032 | Notice | The IPv6 pool "\$pool6_name" was successfully added. |
| 38034 | Notice | In the space "\$site_name", the pool "\$pool_name" was successfully resized (former range: "\$old_start_addr-\$old_end_addr", current range: "\$start_addr-\$end_addr"). |
| 38035 | Error | Unable to resize the pool "\$pool_name" in the space "\$site_name": it includes or excludes already used IP addresses [former range of addresses: "\$old_start_addr-\$old_end_addr", requested range: "\$start_addr-\$end_addr"]. |
| 38036 | Error | Unable to resize the pool "\$pool_name" (\$old_start_addr-\$old_end_addr) in the space "\$site_name": the requested range of addresses (\$start_addr-\$end_addr) is outside the network (\$subnet_start_addr-\$subnet_end_addr). |
| 38039 | Notice | The IPv4 alias "\$alias_name" was successfully deleted (type: \$ip_name_type, address: \$hostaddr, space: \$site_name). |
| 38040 | Notice | The IPv6 alias "\$alias_name" was successfully deleted (type: \$ip6_name_type, address: \$hostaddr, space: \$site_name). |
| 38041 | Notice | The IPv4 alias "\$alias_name" was successfully added (type: \$ip_name_type, address: \$hostaddr, space: \$site_name). |
| 38042 | Notice | The IPv6 alias "\$alias_name" was successfully added (type: \$ip6_name_type, address: \$hostaddr, space: \$site_name). |
| 38100 | Warning | Errors preventing the migration of the network. |
| 38101 | Warning | Unable to add the gateway addresses: the "Gateway offset" defined is outside the new networks. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 40000 | Error | There is not enough space to add this file. |
| 40001 | Error | Max limit reached on this item. |
| 40002 | Error | Selected file exceeds the limit of (\$param). |
| 45000 | Notice | The device "\$hostdev_name" was successfully deleted. |
| 45001 | Notice | The device "\$hostdev_name" was successfully added. |
| 45002 | Notice | The device "\$hostdev_name" was successfully edited. |
| 45003 | Notice | The interface "\$hostiface_name" (\$hostiface_mac) was successfully deleted. |
| 45004 | Notice | The interface "\$hostiface_name" was successfully added. |
| 45005 | Notice | Interface modified. |
| 45008 | Error | Device doesn't exist. |
| 45009 | Error | Interface doesn't exist. |
| 45012 | Error | The device "\$hostdev_name" already exists. |
| 45013 | Error | Interface already exists. |
| 45015 | Error | The device name "\$hostdev_name" already exists. |
| 45016 | Error | The MAC address "\$hostiface_mac" already exists. |
| 45017 | Notice | Device imported. |
| 45018 | Notice | Interface imported. |
| 45020 | Notice | Device name modified. |
| 45021 | Error | Invalid MAC address. |
| 45022 | Notice | Interface managed. |
| 45023 | Notice | Interface unmanaged. |
| 45024 | Warning | The MAC address "\$mac_addr" could not be imported. |
| 45025 | Error | The ports and interfaces are already linked. |
| 45026 | Error | Impossible to unmanage the device \$hostdev_name, an IP address links it with the IPAM. |
| 45027 | Error | Impossible to unmanage the interface \$hostiface_name, an IP address links it with the IPAM. |
| 45028 | Error | Unable to merge the devices "\$hostdev_name" and "\$tohostdev_name": at least one interface links them. |
| 45029 | Error | Unable to link the interface "\$iface1_name" (\$dev1_name) with the 'interface' "\$iface2_name" (\$dev2_name). |
| 50001 | Error | The IP address overlaps another IP address. |
| 50002 | Error | Virtual IP is not contain in any interface. |
| 50003 | Error | IP address already used on interface. |
| 50004 | Notice | SNMP community modified. |
| 50005 | Notice | Ports configuration modified. |
| 50006 | Notice | Traps configuration modified. |
| 50007 | Notice | Date modified. |
| 50008 | Notice | SMTP configuration modified. |
| 50009 | Notice | Versions modified. |
| 50010 | Notice | Firewall configuration modified. |
| 50011 | Notice | SNMP v3 configuration modified. |
| 50012 | Notice | Virtual IP configuration modified. |
| 50013 | Notice | Global configuration modified. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 50014 | Notice | SSH password modified. |
| 50015 | Notice | Static route modified. |
| 50016 | Notice | Interface modified. |
| 50072 | Error | SSH password too short. |
| 50073 | Error | The specified passwords are not identical. |
| 50085 | Notice | The status of the firewall has been modified. |
| 50088 | Notice | Gateway modified. |
| 50090 | Notice | Hostname modified. |
| 50091 | Notice | DNS resolvers modified. |
| 50092 | Notice | SmartHost relay modified. |
| 50111 | Error | You can't delete a certificate which is used by a service (service: \$object_name). |
| 50112 | Notice | SNMP configuration modified. |
| 50113 | Error | Can't create a virtual interface with no physical interfaces attached to it. |
| 50114 | Notice | NTP updated the date of the server using the NTP located at \$server. |
| 50115 | Warning | NTP was not able to synchronize with \$server (message: \$message). |
| 50116 | Error | No suitable servers were found. Could not update the date of the system. |
| 50117 | Error | The service \$service is disabled, cannot start it. |
| 50138 | Error | Unable to delete the SNMP profile "\$snmp_profile_name", it is used by \$nb_devices network device(s): \$iplnetdev_name. |
| 50139 | Notice | Configure DNS Guardian. |
| 50140 | Error | Unable to configure the interface: you cannot set a VIP on its own, at least one other IP address must be set with the VIP service to "None". |
| 51000 | Error | Impossible to edit files that are not in Class Studio format. |
| 51001 | Error | You must be logged in an admin account to edit classes. |
| 51002 | Error | Wrong module name (name = \$module_name). |
| 51003 | Error | Wrong type name (type = \$type_name). |
| 51004 | Error | Wrong path (path = \$path). |
| 51005 | Error | Wrong class name (class = \$class_name). |
| 51017 | Error | You can not add class using the name "global" or "default" (reserved name). |
| 51018 | Error | You can not delete class using the name "global" or "default" (reserved class). |
| 51019 | Error | The name of class file is already used. |
| 51020 | Error | Class file can not be disabled. |
| 51021 | Error | Moving "default" or "global" class files is forbidden. |
| 51022 | Error | Class file can not be enabled - Already enabled. |
| 51023 | Error | Could not find class "\$class_name", please ensure that this class file exists. |
| 51024 | Error | You cannot duplicate the classes "global" and "default". |
| 51202 | Notice | Service successfully enabled. |
| 51203 | Notice | Service successfully disabled. |
| 51204 | Notice | Service successfully started. |
| 51205 | Notice | Service successfully stopped. |
| 51208 | Notice | Configuration file successfully exported to "\$destination". |
| 51209 | Notice | Configuration pushed. |
| 51210 | Error | The remote SOLIDserver could not receive the configuration. |
| 51211 | Notice | Configuration successfully dumped. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 51212 | Error | Unable to update the interface: it is configured by DHCP. |
| 51213 | Error | The VIF does not exist. |
| 51214 | Error | The interface does not exist. |
| 51215 | Error | The IP address is not configured on this interface. |
| 51216 | Error | The interface is already used by another VIF. |
| 51217 | Notice | Configuration successfully imported. |
| 51218 | Error | Unable to change the service status: you cannot change a service status if changes are pending. |
| 51219 | Error | Unable to configure the routing: both configurations must either be filled or empty. |
| 51220 | Error | Unable to configure the routing: the vtysh syntax in command "\$cmd" is invalid. |
| 52000 | Error | Could not connect to the Microsoft service. Please check the IP address and the port the service is listening on. |
| 52001 | Error | The Microsoft server didn't provide the expected information (protocol = \$protocol ; boot time = \$boot_time). |
| 52002 | Error | Could not create the given DNS resource record (name: \$rr_name). |
| 52003 | Error | Could not delete the specified DNS resource record (name: \$rr_name). |
| 52004 | Error | Could not update the options of the Microsoft DNS server (operation that failed: \$operation_name). |
| 52005 | Error | Wrong or missing credentials. |
| 52020 | Error | Could not connect to the WinDNS service (using SSL:\$ssl // Server:\$host_addr:\$port // Error: "\$errmsg"). |
| 52021 | Error | Could not gather data of the server. |
| 52022 | Error | DNS server not referenced (id = \$dns_id). |
| 52023 | Error | Missing or invalid DNS zone ID. |
| 52024 | Error | Could not delete DNS zone "\$dnszone_name". |
| 52025 | Error | Could not retrieve zones list: \$errmsg. |
| 52026 | Error | Couldn't synchronize the zone "\$dnszone_name". Error: \$errmsg. |
| 52027 | Error | Could not create zone "\$dnszone_name". Error: \$errmsg. |
| 52028 | Error | Unsupported zone type. |
| 52040 | Warning | Could not retrieve statistics from Microsoft service. |
| 52041 | Warning | Could not retrieve Microsoft version (check the connection parameters). |
| 52042 | Warning | Could not retrieve the information of the zone \$dnszone_name. |
| 52050 | Notice | The RR "\$rr_name" was successfully added. |
| 52051 | Notice | The RR "\$rr_name" was successfully deleted. |
| 52052 | Notice | Updated Microsoft server configuration (recursion: \$recursion // global forwarders: \$forwarders). |
| 52053 | Notice | DNS zones up-to-date. |
| 52054 | Notice | The DNS zone "\$dnszone_name" was successfully deleted. |
| 52055 | Notice | The DNS zone "\$dnszone_name" was successfully synchronized. |
| 52056 | Notice | Microsoft server updated to version "\$version". |
| 52057 | Notice | The Microsoft server is up and running. |
| 52058 | Notice | Synchronization completed. |
| 52100 | Error | Could not connect to server \$host_addr:\$port (using SSL: \$ssl) . Error message: \$errmsg. |
| 52101 | Error | Could not get the status of the server. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 52102 | Error | Generic synchronization error (line received: \$text). |
| 52103 | Error | The lease timestamp is incorrect: the lease will be ignored. |
| 52125 | Warning | Server option not properly decoded:\ncode = \$option_code\nname = \$option_name\n\$type = \$option_type\nMS option = \$ms_to_ms_option_value\nIPM option = \$ms_to_ipm_option_value\nVendor option = \$option_vendor. |
| 52126 | Warning | Could not find option \$dhcption_name. |
| 52127 | Warning | Could not find scope option \$dhcption_name. |
| 52128 | Warning | Unable to find the static DHCP option "\$dhcption_name". |
| 52150 | Notice | Updated WinDHCP version to \$version. |
| 52151 | Notice | Added \$obj_name (\$obj_type) from vendor \$vendor to the server option definitions. |
| 52152 | Notice | Deleted the static \$hostname because it has no IP associated with it. |
| 52153 | Notice | Flushing delayed delete IP. |
| 52154 | Notice | The DHCP server (\$host_addr) is synchronized (boot time = \$boot_time // serial = \$serial // leases = \$lease_counter). |
| 52155 | Notice | Server option \$dhcption_name deleted. |
| 52156 | Notice | Server option \$dhcption_name modified on SOLIDserver ("\$ms_to_ms_option_value" to "\$ipm_to_ms_option_value"). |
| 52157 | Notice | Server option \$dhcption_name modified on MS ("\$ipm_to_ipm_option_value" to "\$ms_to_ipm_option_value"). |
| 52158 | Notice | Created server option \$dhcption_name on MS (value="\$ipm_to_ms_option_value"). |
| 52159 | Notice | Server option \$dhcption_name ("\$ms_to_ipm_option_value") created in SOLID-server. |
| 52160 | Notice | Server option \$dhcption_name deleted in SOLIDserver. |
| 52161 | Notice | Scope \$dhcpscope_net_addr modified in MS. |
| 52162 | Notice | Scope \$dhcpscope_net_addr deleted in MS. |
| 52163 | Notice | Scope \$dhcpscope_net_addr modified in SOLIDserver. |
| 52164 | Notice | Scope \$dhcpscope_net_addr created in SOLIDserver. |
| 52165 | Notice | Scope \$dhcpscope_net_addr created in MS. |
| 52166 | Notice | Scope \$dhcpscope_net_addr deleted in SOLIDserver. |
| 52167 | Notice | Delete scope option \$dhcption_name on \$dhcpscope_net_addr in MS. |
| 52168 | Notice | Scope option \$dhcption_name modified on \$dhcpscope_net_addr ("\$ipm_to_ms_option_value") in MS. |
| 52169 | Notice | Scope option \$dhcption_name modified on \$dhcpscope_net_addr in SOLIDserver ("\$ipm_to_ipm_option_value" to "\$ipm_to_ms_option_value"). |
| 52170 | Notice | Scope option \$dhcption_name ("\$ms_to_ipm_option_value") created on \$dhcpscope_net_addr in SOLIDserver. |
| 52171 | Notice | Scope option \$dhcption_code created on \$dhcpscope_net_addr in MS. |
| 52172 | Notice | Scope option \$dhcption_code deleted on \$dhcpscope_net_addr in SOLIDserver. |
| 52173 | Notice | Range \$range_start_ip_addr - \$range_end_ip_addr modified in scope \$dhcpscope_net_addr in MS. |
| 52174 | Notice | Range \$range_start_ip_addr - \$range_end_ip_addr deleted in scope \$dhcpscope_net_addr in MS. |
| 52175 | Notice | Range \$range_start_ip_addr - \$range_end_ip_addr deleted from scope \$dhcpscope_net_addr in SOLIDserver. |
| 52176 | Notice | Range \$range_start_ip_addr - \$range_end_ip_addr created in scope \$dhcpscope_net_addr in SOLIDserver. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 52177 | Notice | Range \$range_start_ip_addr - \$range_end_ip_addr created in scope \$dhcp_scope_net_addr in MS. |
| 52178 | Notice | Static \$static_addr modified in MS. |
| 52179 | Notice | Static \$static_addr modified in SOLIDserver. |
| 52180 | Notice | Static \$static_addr deleted in MS. |
| 52181 | Notice | Static \$static_addr deleted in SOLIDserver. |
| 52182 | Notice | Static \$static_addr created in SOLIDserver. |
| 52183 | Notice | Static \$static_addr created in MS. |
| 52184 | Notice | Static option \$dhcption_name deleted on \$static_addr in MS. |
| 52185 | Notice | Static option \$dhcption_code deleted on \$static_addr in SOLIDserver. |
| 52186 | Notice | Static option \$dhcption_name modified ("\$ipm_to_ms_option_value") on \$static_addr in MS. |
| 52187 | Notice | Static option \$dhcption_name modified on \$static_addr in SOLIDserver ("\$ipm_to_ipm_option_value" to "ms_to_ipm_option_value"). |
| 52188 | Notice | Static option \$dhcption_name created ("\$ms_to_ipm_option_value") on \$static_addr in SOLIDserver. |
| 52189 | Notice | Static option \$dhcption_name created ("\$ipm_to_ms_option_value") on \$static_addr in MS. |
| 52200 | Error | Could not locate snmpwalk binary. |
| 52201 | Error | Could not execute snmpwalk command ("\$cmd"). Error: \$err. |
| 52202 | Error | Could not launch tcpdump process (command: "\$cmd"). |
| 52220 | Notice | Dump finished. |
| 52221 | Notice | Captured info about VLAN \$vlan. |
| 52222 | Notice | The snmpwalk was sent using the context "\$context". |
| 52223 | Error | The snmpwalk was successfully executed (\$cmd). |
| 52300 | Error | Could not connect to the remote Microsoft server (IP: \$ip). |
| 53000 | Error | The minimum encryption size is 512 bits. |
| 53001 | Error | Could not generate DSA parameters for the key. |
| 53002 | Error | Could not generate a DSA key. |
| 53003 | Error | Could not generate the CSR file. |
| 53004 | Error | A key file is missing in the imported archive. |
| 53005 | Error | The selected SSL certificate is not usable. |
| 53006 | Error | Unable to change the SSL certificate: invalid certificate (configuration rolled back). |
| 53007 | Error | The import was cancelled: the private key is invalid. |
| 53008 | Error | The import was cancelled: the certificate is invalid. |
| 53009 | Error | The import was cancelled: the private key cannot be used with this certificate. |
| 53010 | Error | Unable to create the SSL object: a Subject Alternative Name (SAN) is invalid. |
| 53011 | Error | Unable to create or import "\$object_name": another SSL object already uses that name. |
| 53012 | Error | Unable to generate the key: its size is invalid (\$size). |
| 53013 | Error | Unable to generate the key: the operation failed. |
| 53014 | Error | Unable to perform this operation: the key cannot be found. |
| 53015 | Error | Unable to delete the key "\$object_name": it is active. |
| 53016 | Error | Unable to delete the key "\$object_name". |
| 53017 | Error | Unable to deactivate the database encryption. |

| Code | Level | Description |
|-------------|--------------|---|
| 53018 | Error | Unable to deactivate the database encryption: it is already deactivated. |
| 53019 | Error | Unable to encrypt the database using the key "\$object_name": the activation failed. |
| 53020 | Error | Unable to activate the key "\$object_name": it is already active. |
| 53021 | Error | Unable to activate the key "\$object_name": it must be saved before being activated. |
| 53022 | Error | Unable to activate the key "\$object_name": it is missing. |
| 53023 | Error | Unable to import the selected key file. |
| 53024 | Error | Unable to activate the database encryption: you must download the key file, saved it in a safe place and compared it to the MD5 checksum. |
| 53025 | Error | Unable to change the database encryption: the database is in read-only. |
| 53026 | Error | Unknown cipher (\$cipher). |
| 53200 | Notice | The DNSSEC key "\$object_name" was successfully deleted. |
| 53201 | Notice | The private key was successfully added. |
| 53202 | Notice | The CSR was successfully added. |
| 53203 | Notice | The public key was successfully added. |
| 53204 | Notice | The self signed certificate was successfully added. |
| 53205 | Notice | Certificate applied. |
| 53206 | Notice | The key \$key_name was successfully invalidated. |
| 53207 | Notice | The key "\$object_name" was successfully generated. |
| 53208 | Notice | The key "\$object_name" was successfully deleted. |
| 53209 | Notice | The database encryption was successfully deactivated. |
| 53210 | Notice | The key "\$object_name" was successfully activated. |
| 53211 | Notice | The selected key file was successfully imported. |
| 53212 | Notice | The SSH key "\$object_name" was successfully deleted. |
| 53213 | Notice | The SSH key "\$key_name" was successfully added. |
| 53250 | Error | Could not generate requested DNSSEC key. Following output was sent: \$parameters. |
| 53251 | Error | Could not fetch key content (\$parameters). |
| 53252 | Error | Could not find DNSSEC key !. |
| 53253 | Error | Could not update key metadata, key malformed. |
| 53254 | Error | Could not find the corresponding key. |
| 53255 | Error | Cannot edit / delete DNSSEC keys. |
| 53256 | Error | Cannot sign the zone "\$dnszone_name", only master zones can be signed. |
| 53257 | Error | Key \$key_name was already revoked. |
| 53258 | Error | The key \$key_name has already been deactivated. |
| 53259 | Error | The key \$key_name has already been activated. |
| 53260 | Error | This key \$key_name was invalidated. Please disable it first before enabling the new one. |
| 53261 | Error | Cannot revoke the key "\$key_name", it is currently disabled. |
| 53262 | Error | Cannot purge the zone "\$dnszone_name", it is currently used for DNSSEC. |
| 53263 | Error | The specified DNSKEY already exists. |
| 53264 | Error | The zone \$dnszone_name was already signed for DNSSEC. |
| 53265 | Error | Cannot change DNS server type, this server currently serves DNSSEC signed zones. |
| 53266 | Error | Cannot change DNS server type, this server is currently configured to resolve DNSSEC zones. |
| 53267 | Error | Cannot directly delete DNSSEC-related records. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 53268 | Error | Cannot delete last anchor, at least 1 must be specified. |
| 53269 | Error | Cannot delete "\$object_name", it is currently used by the system. |
| 53270 | Error | It's impossible to delete KSKs. Please use the options of the Expert menu instead. |
| 53271 | Error | You must select at least 1 DNSSEC key. |
| 53272 | Error | Cannot invalidate the key "\$key_name", only KSK keys can be invalidated. |
| 53273 | Error | Cannot enable the out-of-date key "\$key_name". |
| 53274 | Error | Cannot add Trust Anchor from invalidated KSK. |
| 53275 | Error | You can ONLY enable/disable KSK or ZSK keys. |
| 53276 | Error | This DNS server (\$dns_name) is already refreshing. |
| 53277 | Error | This action cannot be performed on non-signed DNS zones (zone: \$dnszone_name). |
| 53278 | Error | The minimal allowed validity for DNSSEC keys is 3 days. |
| 53280 | Error | You cannot set the validity to \$count days. The date range cannot exceed the year 2037. |
| 53281 | Error | Invalid GSS-TSIG key. |
| 53282 | Error | Unable to set the update-policy on the zone "\$dnszone_name" (type: \$dnszone_type) on the server "\$dns_name": no key was found. |
| 53283 | Error | Unable to add the DS record: there is no delegation record (NS) with the same name in the zone. |
| 53284 | Error | Unable to revoke the \$key_type \$key_name: only ZSK can be revoked. |
| 53285 | Error | The DNS key does not exist. |
| 53288 | Error | Unable to generate the DNSSEC \$key_type key for "\$zone_name": the value of "\$parameters" exceeds the maximum UNIX timestamp. |
| 53289 | Warning | To avoid exceeding the maximum UNIX timestamp, the \$key_type key validity period has been reduced for the zone "\$zone_name". |
| 53290 | Error | Unable to execute the rollover of the KSK "\$key_name": it has been revoked. |
| 53291 | Error | Unable to revoke the KSK "\$key_name" it is the only one protecting the zone: you must first execute its rollover or generate a new one. |
| 53292 | Error | Unable to add the Trust Anchor: its owner "\$owner" does not comply with RFC 1034. |
| 53293 | Error | Unable to add the SSH key "\$key_name". |
| 53294 | Error | Unable to delete the SSH key "\$object_name": it is used by the DHCP server "\$dhcp_name". |
| 53301 | Error | Unsupported advanced property. |
| 53302 | Error | TSIG keys are not supported. |
| 53303 | Error | Unable to set the recursion parameter on the server "\$dns_name" as long as the box "Force Hybrid DNS compatibility" is ticked. |
| 53304 | Error | Unable to add the view: you cannot add views on a Hybrid compliant DNS server. |
| 53305 | Error | Unable to add the zone "\$dnszone_name": you cannot add "\$dnszone_type" zones on this type of architecture. |
| 53306 | Error | Unable to add the "\$dnszone_type" zone "\$dnszone_name": you cannot manage authoritative zones and recursive zones on this type of architecture. |
| 53307 | Error | Unable to add the record "\$rr_name": you cannot add "\$rr_type" records on this type of architecture. |
| 53308 | Error | The operation could not complete on the smart server "\$dns_name": it is not supported by at least one of the physical servers managed through the smart. |
| 53309 | Error | The DNS server "\$dns_name" must be managed through a smart architecture. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 53310 | Error | Unable to switch to NSD/Unbound: the server configuration of "\$dns_name" is not compatible with Hybrid DNS. To generate the list of all the parameters that prevent the switch, use the menu: Report > Hybrid DNS Engine incompatibilities. |
| 53311 | Error | Unable to switch "\$dns_name" to \$destination: its engine cannot be switched to Hybrid DNS. |
| 53312 | Error | Unable to set the parameter "forward" in the zone "\$dnszone_name": it is not supported on stub zones or Hybrid configurations. |
| 53313 | Error | Unable to set the parameter "forwarders" on the zone "\$dnszone_name": it is not supported on stub zones or Hybrid configurations. |
| 53314 | Error | Unable to disable the Force Hybrid DNS compatibility option: you cannot disable it when the smart architecture is managing at least one Hybrid engine. |
| 53315 | Error | You must switch your Hybrid engine to BIND before adding them to a smart architecture. |
| 53316 | Error | Unable to add the zone: you cannot add a "\$dnszone_type" zone as long as the server recursion parameter is set to "\$recursion". |
| 53317 | Error | Unable to set forward related options on the server "\$dns_name": you cannot configure the forward or set forwarders on a server with no recursion if the box "Force Hybrid DNS compatibility" is ticked. |
| 53318 | Error | Unable to switch the server "\$dns_name": you can only switch physical servers that are managed through a smart architecture. |
| 53319 | Error | Unable to perform this operation: you cannot set forwarding parameters on authoritative Hybrid architectures. |
| 53320 | Error | Unable to switch the engine of the DNS server "\$dns_name". |
| 53322 | Error | Unable to sign the zone: DNSSEC is incompatible with Hybrid servers. |
| 53323 | Error | Forcing Hybrid DNS compatibility is only possible on smart architectures. |
| 53324 | Error | Unable to add the record "\$rr_name": its value contains \$rr_value_length characters. It cannot exceed 4000 characters. |
| 53330 | Notice | Unable to switch the server "\$dns_name" to "\$switch_to": you are currently using this configuration. The server was not edited. |
| 53331 | Notice | The server "\$dns_name" DNS service configuration was successfully switched. |
| 54501 | Error | Unable to add the connection profile: it already exists. |
| 54502 | Error | Unable to delete the connection profile "\$profile_name": it does not exist. |
| 54503 | Error | Unable to use the connection profile name "\$profile_name": it already exists. |
| 54504 | Error | Unable to delete the connection profile "\$profile_name": it is currently used. |
| 54510 | Notice | The connection profile "\$profile_name" was successfully added. |
| 54511 | Notice | The connection profile was successfully deleted. |
| 55000 | Warning | Some conflict detected. |
| 55001 | Warning | To commit use \$cmd. |
| 57001 | Error | Unable to perform the command "force \$rndc_cmd" for the zone "\$dnszone_name": it is not supported on "\$dnszone_type" zones. |
| 57100 | Notice | The directory was successfully synchronized. |
| 57101 | Notice | The directory was successfully deleted. |
| 57102 | Error | Unable to add a directory: it already exists. |
| 57103 | Error | Unable to delete the directory: it was not found. |
| 57108 | Notice | On IP address "\$hostaddr", a session was found for "\$identity" in the directory "\$directory". |
| 57109 | Notice | No session found. |
| 58002 | Error | Unable to restrict the class parameter "\$tag_name": it is already propagated. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 58003 | Error | Unable to inherit the class parameter "\$tag_name": you are at the highest level of hierarchy you can only propagate parameters. |
| 58004 | Error | Unable to inherit the class parameter "\$tag_name": the parameter is restricted in the parent object. |
| 58005 | Error | Unable to inherit the class parameter "\$tag_name": it was not found in the parent object. |
| 58006 | Error | Unable to delete the class parameter "\$tag_name": you cannot remove an inherited parameter. |
| 58007 | Error | Unable to add/edit the resource because of the class parameter "\$tag_name": an inherited class parameter cannot be restricted in template mode. |
| 58008 | Error | Unable to perform this operation on the VLSM block-type network \$subnet_addr-\$subnet_end_addr (space "\$site_name"): it can only be executed if the value Inheritance property is "Inherit", but the class parameter "\$tag_name" is "Set". |
| 58009 | Error | Unable to add/edit the resource: the default class parameter "\$tag_name" cannot be removed. |
| 58011 | Error | Unable to set the advanced property "\$advanced_property": permission denied. |
| 58012 | Error | Unable to set the advanced property Default domain to "\$domain": its value is not part of the Selected domains list "\$domain_list". |
| 58013 | Error | Unable to perform the replication: you are not granted the right "\$service_name". |
| 58014 | Error | Unable to add the IP address with the advanced property Shortname to "\$hostname": it requires either disabling the DHCP replication or enabling Use IPAM name instead of DHCP client name [use_ipam_name=1]. |
| 58015 | Error | Unable to add the IP address: the value of the properties Shortname "\$hostname" and Default domain "\$domain" does not match the full IP address name "\$ip_name". |
| 58016 | Error | Unable to set the advanced property DNS view for reverse zones to "\$rev_dns_view_name": it requires setting a DNS server for reverse zones [rev_dns_name]. |
| 58017 | Error | Unable to set the advanced property DNS view to "\$dns_view_name": it requires setting a DNS server [dns_name]. |
| 58018 | Error | Unable to set the advanced property Corresponding IPv6 address "\$ipv6_mapping_address": it requires enabling Activate the IPv4 to IPv6 transition [ipv6_mapping=1]. |
| 58019 | Error | Unable to set the advanced property IPv6 network (subnet) "\$ipv6_mapping_subnet": it requires enabling Activate the IPv4 to IPv6 transition [ipv6_mapping=1]. |
| 58020 | Error | Unable to rename the IPv6 address replicated in the DNS: this editing should delete the matching AAAA record but it is not part of your resources. |
| 58021 | Error | Unable to rename the IPv4 address replicated in the DNS: this editing should delete the matching A record but it is not part of your resources. |
| 58022 | Error | Unable to rename the IPv4 address with alias replicated in the DNS: this editing should delete the matching CNAME record but it is not part of your resources. |
| 58023 | Error | Unable to rename the IP address with alias replicated in the DNS: this editing should add the matching CNAME record but it is not part of your resources. |
| 58024 | Error | Unable to rename the IPv6 address replicated in the DHCP: this editing should delete the matching static but it is not part of your resources. |
| 58025 | Error | Unable to rename the IPv4 address replicated in the DHCP: this editing should delete the matching static but it is not part of your resources. |
| 58026 | Error | Unable to add the IPv6 address and replicate it in the DHCP: none of the parent objects of the matching static are part of your resources. |
| 58027 | Error | Unable to add the IPv4 address and replicate it in the DHCP: none of the parent objects of the matching static are part of your resources. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 58029 | Error | Unable to add the IPv4 address and replicate it in the DNS: this operation should add the matching A but there is a conflict. Another record of the same name already exists but as it is not part of your resources you cannot overwrite it. |
| 58030 | Error | Unable to add the IPv6 address and replicate it in the DNS: none of the parent objects of the matching AAAA record are part of your resources. |
| 58031 | Error | Unable to add the IPv4 address and replicate it in the DNS: none of the parent objects of the matching A record are part of your resources. |
| 58032 | Error | Unable to add the IP address and replicate it in the DNS: this operation should add the matching PTR but there is a conflict. Another record with the same name already exists but as it is not part of your resources you cannot overwrite it. |
| 58033 | Error | Unable to add the IP address and replicate it in the DNS: this operation should add the matching PTR but there is a conflict. Another record with the same name already exists but as it is not part of your resources you cannot overwrite it. |
| 58034 | Error | Unable to add the IP address and replicate it in the DNS: none of the parent objects of the matching PTR record are part of your resources. |
| 58036 | Error | Unable to rename the IPv4 address and stop its replication in the DNS: this editing should delete the matching A record but it is not part of your resources. |
| 58037 | Error | Unable to edit the IP address and stop its replication in the DNS: this editing should delete the matching PTR record but it is not part of your resources. |
| 58038 | Error | Unable to add the IPv4 address and activate the transition to IPv6: none of the parent objects of the matching IPv6 address are part of your resources. |
| 58039 | Error | Unable to edit the IPv4 address and deactivate the transition to IPv6: this editing should delete the matching IPv6 address but it is not part of your resources. |
| 58040 | Error | Unable to edit the IPv6 pool and stop its replication in the DHCP: this editing should delete the matching range but it is not part of your resources. |
| 58041 | Error | Unable to edit the IPv4 pool and stop its replication in the DHCP: this editing should delete the matching range but it is not part of your resources. |
| 58042 | Error | Unable to add the IPv6 pool and replicate it in the DHCP: none of the parent objects of the matching range are part of your resources. |
| 58043 | Error | Unable to add the IPv4 pool and replicate it in the DHCP: none of the parent objects of the matching range are part of your resources. |
| 58044 | Error | Unable to edit the IPv6 network and stop its replication in the DHCP: this editing should delete the matching scope but it is not part of your resources. |
| 58045 | Error | Unable to edit the IPv4 network and stop its replication in the DHCP: this editing should delete the matching scope but it is not part of your resources. |
| 58046 | Error | Unable to add the IPv6 network and replicate it in the DHCP: none of the parent objects of the matching scope are part of your resources. |
| 58047 | Error | Unable to add the IPv4 network and replicate it in the DHCP: none of the parent objects of the matching scope are part of your resources. |
| 58048 | Error | Unable to edit the network and stop its replication in the DNS: this editing should delete the matching reverse zone but it is not part of your resources. |
| 58049 | Error | Unable to add the network and replicate it in the DNS: none of the parent objects of the matching reverse zone are part of your resources. |
| 58050 | Error | Unable to add the IPv4 network and activate the transition to IPv6: none of the parent objects of the matching IPv6 network are part of your resources. |
| 58051 | Error | Unable to edit the IPv4 network and deactivate the transition to IPv6: this editing should delete the matching IPv6 network but it is not part of your resources. |
| 59001 | Notice | The application "\$application_name" was successfully added. |
| 59002 | Notice | The pool "\$pool_name" was successfully added. |
| 59003 | Notice | The application "\$appapplication_name" (FQDN: "\$appapplication_fqdn") was successfully deleted. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 59004 | Notice | The pool "\$apppool_name" was successfully deleted. |
| 59005 | Notice | The node "\$appnode_name" was successfully deleted. |
| 59006 | Error | Application already exists (name: \$name, FQDN: \$fqdn). |
| 59007 | Error | Unable to associate the application with the server "\$gslbserver_name": it does not support GSLB. |
| 59008 | Error | Unable to associate the GSLB server "\$gslbserver_name" with the application "\$app_name": the application "\$name" (FQDN: \$fqdn) is already associated with this server. |
| 59011 | Error | Unable to find the application. |
| 59012 | Error | Unable to edit the application: deployed traffic policies are read-only. |
| 59013 | Error | Unable to find the application or pool. |
| 59014 | Error | Unable to add the pool "\$name": it already exists in the application "\$application_name" (FQDN: "\$application_fqdn"). |
| 59015 | Error | Unable to rename the pool: the application already contains a pool named "\$name". |
| 59016 | Error | Unable to add the pool "\$name": the application already contains a pool using this protocol (\$type). |
| 59017 | Error | Unable to find the pool. |
| 59018 | Error | Unable to find the pool or node. |
| 59019 | Error | Unable to add the node "\$name": it already exists in the pool "\$pool_name" of the application "\$application_name" (FQDN: "\$application_fqdn"). |
| 59020 | Error | Unable to rename the node: the pool already contains a node named "\$name". |
| 59023 | Error | Unable to edit the node: deployed traffic policies are read-only. |
| 59024 | Error | Unable to delete the node: deployed traffic policies are read-only. |
| 59025 | Error | Unable to edit the pool: deployed traffic policies are read-only. |
| 59026 | Error | Unable to delete the pool: deployed traffic policies are read-only. |
| 59027 | Error | Unable to delete the deployed application "\$name" (FQDN: "\$fqdn", GSLB server: "\$gslbserver"): it is in read-only. You must either delete the parent application or dissociate it from the relevant GSLB server. |
| 59028 | Error | Unable to find the node. |
| 59029 | Error | Unable to add the node: the IP address "\$hostaddr" already exists in the pool. |
| 59030 | Notice | The node "\$node_name" was successfully added. |
| 59038 | Error | Unable to associate the GSLB server "\$gslbserver" with the application "\$name" (FQDN: "\$fqdn"): you reached the maximum number of applications associated with one server. |
| 59039 | Error | Unable to add the node "\$name" in the pool "\$pool_name": you reached the maximum number of nodes in one pool. |
| 59040 | Error | Unable to delete the DNS server: it is associated with a deployed application traffic policy. Go to the module Application to dissociate the server from the application. |
| 59041 | Notice | The node "\$appnode_name" was successfully managed (pool: \$apppool_name, application: \$appapplication_name). |
| 59042 | Notice | The node "\$appnode_name" was successfully unmanaged (pool: \$apppool_name, application: \$appapplication_name). |
| 59044 | Error | Unable to convert the DNS server "\$dns_name" into a smart architecture: it is associated with a deployed application traffic policy. Go to the module Application to dissociate the server from the application. |
| 59045 | Error | Unable to associate the application with the server "\$gslbserver_name": it does not exist or is not in your resources. |
| 59046 | Error | Unable to delete the application "\$name" (FQDN: \$fqdn): it is associated with a GSLB server that is not part of your resources. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 59047 | Error | Unable to delete the pool "\$pool_name": it belongs to the application "\$application_name" (FQDN: \$application_fqdn) which is associated with a GSLB server that is not part of your resources. |
| 59048 | Error | Unable to delete the node "\$node_name" (pool: \$pool_name): it belongs to the application "\$application_name" (FQDN: \$application_fqdn) which is associated with a GSLB server that is not part of your resources. |
| 59049 | Error | Unable to complete the deletion of the application "\$appapplication_name": you cannot complete the deletion of a parent application. |
| 59050 | Error | Unable to complete the deletion of the application "\$appapplication_name" deployed on the GSLB server "\$appapplication_gslbserver_name": it is not in Delayed delete. |
| 59101 | Notice | No lease inconsistency was found. |
| 59102 | Notice | Some lease inconsistencies were found, to repair them refer to the file \$file (Administration/Local files listing). |
| 59103 | Error | Unable to repair the leases: no failover channel was found. |
| 59104 | Error | Unable to repair the leases: the download of the lease file of the server "\$dhcp_name" (\$hostaddr) was interrupted. |
| 59105 | Error | Unable to repair the leases: the lease file downloaded from the server "\$dhcp_name" is corrupted. |
| 63000 | Error | Unable to add or rename the list template: this name already exists. |
| 63001 | Error | Unable to delete the list template: it does not exist. |
| 63002 | Error | Unable to delete the list template: you cannot delete "default". |
| 63003 | Error | Unable to rename the list template: you cannot rename "default". |
| 63100 | Notice | The list template "\$name" was successfully deleted. |
| 64001 | Notice | The Cloud Observer worker "\$coworker_name" was successfully deleted. |
| 64002 | Error | Unable to find the Cloud Observer plugin. |
| 64003 | Error | Unable to find the Cloud Observer worker. |
| 64004 | Notice | The Cloud Observer worker "\$coworker_name" was successfully enabled. |
| 64005 | Notice | The Cloud Observer worker "\$coworker_name" was successfully disabled. |
| 64006 | Error | Unable to execute the worker "\$name" (\$uuid): it is disabled. |
| 64007 | Error | Unable to execute the worker "\$name" (\$uuid): plugin not found. |
| 64008 | Notice | The worker "\$name" (\$uuid) is being executed. |
| 64009 | Error | Unable to add the plugin "\$coplugin_name": it already exists. |
| 64010 | Error | Unable to add the worker "\$coworker_name": it already exists. |
| 64011 | Error | Unable to add the worker: you cannot specify a UUID, it is generated automatically. |
| 64012 | Notice | The Cloud Observer worker "\$coworker_name" (UUID: \$coworker_uuid) was successfully added. |
| 64013 | Notice | The Cloud Observer worker "\$coworker_name" was successfully edited. |
| 65000 | Notice | "\$object_details" from the module "\$object_origin" was successfully remediated. |
| 65001 | Error | Unable to edit "\$object_details" from the module "\$object_origin" in the IPAM: no IP address to edit was found. |
| 65002 | Error | Unable to add "\$object_details" from the module "\$object_origin" in the IPAM: the IP address already exists in the space "\$ipam_site_name" or one of its child spaces. |
| 65003 | Error | Unable to remediate "\$object_details" from the module "\$object_origin" in the IPAM: no space is configured. |
| 65004 | Error | Unable to remediate "\$object_details" from the module "\$object_origin" in the IPAM: a DHCP lease already matches the corresponding IP address. |
| 65005 | Error | Unable to remediate "\$object_details" from the module "\$object_origin" in the IPAM: a DHCP static already matches the corresponding IP address. |

Return Codes

| Code | Level | Description |
|-------------|--------------|--|
| 65006 | Error | Unable to remediate "\$object_details" from the module "\$object_origin" in the IPAM: the space "\$ipam_site_name", or one of its child spaces, contains no terminal network to receive it. |
| 65007 | Error | Unable to remediate "\$object_details" from the module "\$object_origin" in the IPAM: the space "\$ipam_site_name", or one of its child spaces, contains several terminal networks that can receive it. |
| 65008 | Error | Unable to remediate "\$object_details" from the module "\$object_origin" in the IPAM: its MAC address "\$mac_addr" already matches a DHCP lease. |
| 65009 | Error | Unable to remediate "\$object_details" from the module "\$object_origin" in the IPAM: its MAC address "\$mac_addr" already matches a DHCP static. |
| 65010 | Error | Unable to remediate "\$object_details" from the module "\$object_origin" in the IPAM: the DNS record "\$rr_name" (type: \$rr_type) already exists. |
| 65011 | Error | Unable to remediate "\$object_details" from the module "\$object_origin" in the IPAM: you cannot remediate data if the DNS replication is enabled. |
| 65012 | Error | Unable to remediate "\$object_details" from the module "\$object_origin" in the IPAM: you cannot remediate data if the DHCP replication is enabled. |
| 80001 | Error | Unable to add the dashboard: it already exists. |
| 80002 | Error | Unable to delete the dashboard: it does not exist. |
| 80003 | Error | Unable to delete the dashboard: it is not empty. |
| 80004 | Error | Unable to delete the dashboard: default dashboards cannot be deleted. |
| 80005 | Error | Unable to edit the dashboard: default dashboards cannot be edited. |
| 81001 | Error | \$parameters. |
| 82000 | Error | Unable to add the folder "\$nomfolder_path": it already exists. |
| 82001 | Error | Unable to migrate the folder "\$nomfolder_path": it already exists in this target folder or one of its sub-folders. |
| 82002 | Error | Unable to find the folder. |
| 82003 | Notice | The folder "\$nomfolder_name" was successfully deleted. |
| 82004 | Error | Unable to delete the folder "\$nomfolder_path": it contains at least one subfolder. |
| 82005 | Error | Unable to find the folder or network object. |
| 82006 | Error | Unable to add the network object "\$nomnetobj_name": it already exists in the folder "\$nomfolder_path". |
| 82008 | Error | Unable to find the network object. |
| 82009 | Notice | The network object "\$nomnetobj_name" was successfully deleted. |
| 82010 | Error | Unable to delete the network object "\$nomnetobj_name" from the folder "\$nomfolder_path": it is set as parent of another network object. |
| 82011 | Error | Unable to find the network object or interface. |
| 82012 | Error | Unable to add the interface "\$port_name - \$nomiface_vlan_number - \$nomiface_hostaddr": it already exists in the network object "\$nomnetobj_name" of the folder "\$nomfolder_path". |
| 82013 | Error | Unable to find the interface. |
| 82014 | Notice | The interface "\$nomiface_fullname" was successfully deleted. |
| 82015 | Error | The interface "\$port_name - \$nomiface_vlan_number - \$nomiface_name - \$nomiface_ip_addr" was added but it cannot be set as main IPv4 interface: the network object "\$nomnetobj_name" of the folder "\$nomfolder_path" already has "\$main_ip" as main interface. |
| 82016 | Error | The interface "\$port_name - \$nomiface_vlan_number - \$nomiface_name - \$nomiface_ip_addr" was added but it cannot be set as main IPv6 interface: the network object "\$nomnetobj_name" of the folder "\$nomfolder_path" already has "\$main_ip" as main interface. |

Return Codes

| Code | Level | Description |
|-------------|--------------|---|
| 82017 | Error | Unable to connect the port "\$port_name" (network object: \$nomnetobj_name, folder: \$nomfolder_path) to the port "\$connected_port_name" (network object: \$connected_port_nomnetobj_name, folder: \$connected_port_nomfolder_path): you cannot connect a port to an already connected port. |
| 82018 | Notice | The folder "\$nomfolder_name" was successfully migrated. |
| 82019 | Notice | The network object "\$nomnetobj_name" was successfully edited. |
| 82020 | Notice | The folder "\$nomfolder_name" was successfully added. |
| 82021 | Notice | The network object "\$nomnetobj_name" was successfully added. |
| 82022 | Notice | The interface "\$port_name" was successfully added. |
| 82023 | Notice | The folder was successfully imported. |
| 82024 | Notice | The network object was successfully imported. |
| 82025 | Notice | The interface was successfully imported. |
| 82026 | Error | Unable to add the interface port "\$port_name" with the MAC address "\$port_mac" in the network object "\$nomnetobj_name"(folder: \$nomfolder_path): this address is already used by another interface port. |
| 82027 | Warning | The raw import was performed but the port of the interface "\$nomiface_port - \$nomiface_vlan_number - \$nomiface_hostaddr" cannot be connected (network object: \$nomnetobj_name, folder: \$nomfolder_path): the port it should be connected to does not exist. |
| 82028 | Notice | The network object "\$nomnetobj_name" was successfully migrated. |
| 82029 | Error | Unable to set the interface "\$port_name - \$nomiface_vlan_number - \$nomiface_name" as main interface of the network object "\$nomnetobj_name" (folder: \$nomfolder_path): a main interface must have an IP address. |
| 82030 | Notice | The ports "\$port_name" and "\$connected_port_name" were successfully connected. |
| 82031 | Error | Unable to connect the port "\$port_name" : you cannot connect a port to itself. |
| 82032 | Warning | The raw import was performed but the network object "\$nomnetobj_name" (folder: "\$nomfolder_path") cannot be associated with its parent: "\$parent_nomnetobj_name" (folder: \$parent_nomfolder_path) was not found. |
| 82033 | Notice | The interface "\$port_name - \$vlan_id - \$nomiface_name - \$nomiface_hostaddr" was successfully added. |
| 82034 | Error | Unable to add the network object "\$nomnetobj_name" in the folder "\$nomfolder_path": you reached the maximum number of network objects (\$max). |
| 82035 | Error | Unable to edit this folder: you cannot edit parent folders, your rights only apply to child folders. |
| 82036 | Error | Unable to add this folder: you cannot add parent folders, your rights only apply to child folders. |
| 82037 | Error | Unable to delete this folder: you cannot delete parent folders, your rights only apply to child folders. |