Brocade® Fabric OS® REST API Reference Manual, 9.2.x

Reference Manual
April 28, 2023

Table of Contents

Introduction **29**
Supported Hardware and Software **29**
Brocade REST API Courses **29**
REST Terminology **29**
Terms Used in YANG Files 31
Standards and Related References **32**
Contacting Technical Support for Your Brocade® Product **34**
Document Feedback **35**
Fabric OS REST API Overview **36**
FOS REST API Described **36**
Functionality 37
FOS REST API URI Restrictions 38
Security and the FOS REST API 38
Zoning Operations 39
Diagnostic Port Requests 40
Protocol Support 40
Scalability Recommendations for FOS REST API Clients 40
FOS REST API and Brocade Extension 41
Restrictions on FOS REST API Configuration Attributes and Statistics 41
Using the Brocade FOS REST API **42**
Using Brocade FOS REST API Session-Less Operation **45**
FOS REST API Role-Based Access Control **47**
HTTP Header **48**
Request Headers **49**
Media Types **49**
HTTP Status Codes and Messages **49**
Supported Methods **50**
DELETE Method 50
GET Method 50
HEAD Method 52
OPTIONS Method 52
PATCH Method 53
POST Method 54
Query Parameter Support **54**
FOS REST API Version History **55**
FOS REST API and Brocade Virtual Fabrics **58**

Fabric OS REST Session Configuration **59**
Enabling and Disabling the Fabric OS REST Interface 60
Encrypted Authorization Token 60
Generating an Encrypted Authorization Token 60
Enabling and Disabling HTTP Protocol Support 61
Enabling and Disabling Keepalive Mode 62
Configuration of Fabric OS REST Interface Session Values 62
Identification and Termination of Fabric OS REST Sessions 62
YANG Module Overview **63**
Supported Data Types 64
FOS REST API YANG Modules **64**
FOS REST API Modules for Operations **69**
brocade-login-response **69**
brocade-operation-configdownload **71**
brocade-operation-configupload **73**
brocade-operation-date **74**
brocade-operation-device-login-rebalance **75**
brocade-operation-diagnostics **76**
brocade-operation-extension **78**
brocade-operation-fabric **81**
brocade-operation-factory-reset **82**
brocade-operation-firmwaredownload **84**
brocade-operation-license **91**
brocade-operation-lldp **94**
brocade-operation-management-ethernet-interface **96**
brocade-operation-ntp-clock-server **97**
brocade-operation-pcie-health **98**
brocade-operation-port **100**
brocade-operation-portchannel **103**
brocade-operation-port-decommission **104**
brocade-operation-reboot **106**
brocade-operation-restart **107**
brocade-operation-sdd-quarantine **108**
brocade-operation-security-acl-policy **110**
brocade-operation-security-authentication-configuration **112**
brocade-operation-security-authentication-secret **114**
brocade-operation-security-certificate **115**
brocade-operation-security-fabric-wide-policy-distribute **116**
brocade-operation-security-policy-chassis-distribute **117**
brocade-operation-security-policy-distribute **119**

brocade-operation-security-reset-violation-statistics....120
brocade-operation-security-role-clone....121
brocade-operation-show-status....122
brocade-operation-supportlink....124
brocade-operation-supportsave....126
brocade-operation-traffic-optimizer....128
brocade-operation-usb-delete-file....131
brocade-operation-zone....132
FOS REST API Modules for Fibre Channel Features....134
brocade-access-gateway....135
Examples....136
Viewing the Access Gateway Mode....136
Enabling Access Gateway Mode on a Switch....137
Creating a New Port Group....137
Adding an N_Port to a Port Group....138
Adding an F_Port to a Port Group....138
Enabling Load-Balancing Mode on a Port Group....139
Enabling Multiple Fabric Name Monitor Mode on a Port Group....139
Deleting Ports From a Port Group....139
Deleting a Port Group....140
Mapping an F_Port to an N_Port....140
Deleting an F_Port From an N_Port Mapping....141
Viewing End Device Information....141
Configuring an N_Port....143
Configuring the Reliability Counter for N_Ports....143
Enabling Failover for an N_Port....143
Enabling Failback for an N_Port....144
Viewing the List of Devices Logged On to the Access Gateway Switch....144
brocade-application-server....146
Examples....146
Viewing the Application Server Information for an Entity ID....146
Viewing the Application Server Information for an N_Port ID....147
brocade-chassis....148
Examples....150
Viewing the Chassis Name....150
Viewing the Chassis World Wide Name....151
Enabling Virtual Fabrics on the Chassis....151
Viewing the High Availability Status....152
Determining if HA Is Enabled....152
Determining the HA Synchronized Status....153

brocade-extension **154**
Examples 157
Viewing the Extension HCL Status 157
Viewing the Traffic Control Lists 158
Configuring a Traffic Control List 159
Enabling the Admin Status of a Traffic Control List 160
Deleting a Traffic Control List 160
Viewing the LAN Flow Statistics 160
Viewing the Global LAN Statistics 161
brocade-extension-ip-route **164**
Examples 164
Creating an IP Route 164
Deleting an IP Route 165
Retrieving an IP Route 165
brocade-extension-ipsec-policy **167**
Examples 167
Creating an IPsec Policy Using a Preshared Profile 167
Creating an IPsec Policy Using a PKI Profile 167
Deleting an IPsec Policy 168
Retrieving a List of IPsec Policies 168
brocade-extension-tunnel **170**
Examples 175
Creating a Tunnel 175
Creating a Circuit on a Tunnel 175
Deleting a Tunnel 176
Viewing a List of Circuit QoS Statistics 176
Viewing the LAN Flow Statistics 177
brocade-fabric **180**
Examples 180
Retrieving Switch Information 180
Retrieving Switch Information From a Virtual Fabric 181
Viewing Attached Access Gateway Switches 182
brocade-fabric-traffic-controller **184**
Examples 186
Retrieving Fabric Traffic Controller Information 186
Retrieving Fabric Traffic Controller Information for a Specific N_Port 187
brocade-fdmi **190**
Examples 191
Retrieving HBA Port Attributes 191
brocade-fibrechannel-configuration **193**

Examples....194
Configuring the Maximum Number of Logins Allowed on a Port....194
Viewing the Port Configuration....194
Configuring the Dynamic Port Name Format....195
Enabling Node Name Checking for Zoning....195
Modifying the Fabric Lock Timeout....196
Enabling Insistent Domain ID on a Device....196
Configuring the Fabric....196
Configuring the Chassis Settings....197
brocade-fibrechannel-diagnostics....198
Examples....199
Retrieving the Test Status of a Diagnostic Port....199
Starting a Diagnostic Port Test....200
Stopping a Diagnostic Port Test....200
brocade-fibrechannel-logical-switch....201
Examples....201
Retrieving Logical Switch Information....201
Retrieving Information for a Logical Switch Instance....203
Creating a Logical Switch....203
Assigning Ports to a Logical Switch....204
Creating a FICON Logical Switch....205
Deleting a Logical Switch....205
Removing Ports from a Logical Switch....206
Enabling the Base Switch....206
Disabling the Base Switch....207
Enabling FICON Mode on a Logical Switch....207
Enabling a Logical ISL on a Logical Switch....208
Disabling a Logical ISL on a Logical Switch....208
brocade-fibrechannel-routing....209
Examples....210
Configuring FC-FC Routing....210
Setting the Maximum LSAN Count....211
Assigning a Backbone Fabric ID....211
Enabling Shortest IFL Mode....211
Disabling Shortest IFL Mode....212
Configuring Enforce LSAN Tags....212
Configuring a Speed LSAN Tag....213
Deleting an LSAN Tag....213
Viewing the LSAN Zone Information....214
Viewing the LSAN Device Information....215

Configuring an Alias for an Edge Fabric....216
Enabling FC Routing....216
Disabling FC Routing....217
Configuring the Fabric ID for the Edge Fabric....217
Enabling a Port as an EX_Port....217
Configuring the Preferred Domain ID....218
brocade-fibrechannel-switch....219
Examples....220
Retrieving Switch Resource Information....220
Retrieving Switch Resource Information from a Virtual Fabric....221
Setting the User-Friendly Name for a Switch....223
Setting the Login Banner for a Switch....223
brocade-fibrechannel-trunk....224
Examples....225
Viewing Trunk Settings for a Port....225
Enabling Trunking on a Port....225
Viewing Trunk Data for a Source Port....226
Viewing Trunk Performance Data....226
Viewing Performance Data for a Specific Trunk Group....227
Viewing Data for all Trunk Area Groups....228
Configuring a Trunk Area Group....229
Deleting a Trunk Area Group....229
Deleting Members From an Existing Trunk Area Group....230
brocade-ficon....231
Examples....232
Viewing the FICON CUP Configuration....232
Enabling FMS Mode....233
Configuring FICON CUP....234
Viewing the FICON Logical Paths....234
Configuring a FICON Logical Path....235
Viewing Request Node Identifier (RNID) Data....236
Viewing Switch RNID Data....237
Viewing Link Incident Registration Record (LIRR) Data....237
Viewing Registered Link Incident Record (RLIR) Data....238
brocade-firmware....240
Examples....240
Viewing the Firmware Version on a Device....240
Checking the Firmware Integrity Check Status....241
Enabling the Firmware Integrity Check....242
brocade-fru....243

Examples....245
Setting the Extension Application Mode Configuration....245
Setting the Extension VE Mode Configuration....246
Setting the Extension GE Mode Configuration....246
Viewing Data for all Fans in the Chassis....247
Viewing Data for a Specific Fan....248
Viewing Data for a Specific Power Supply....248
Viewing the History Log....249
brocade-interface/extension-ip-interface....253
Examples....253
Creating an IP Interface....253
Deleting an IP Interface....254
Retrieving Information on an IP Interface....254
brocade-interface/fibrechannel....256
Examples....261
Retrieving a List of Logical E_Ports....262
Getting Details on a Single Fibre Channel Port....262
Getting Details on a Single Fibre Channel Port Attribute....264
Configuring a Single Fibre Channel Port Attribute....265
Setting the User-Friendly Name for a Fibre Channel Interface....265
Setting a Fibre Channel Interface State....265
Getting Trunk Settings for a Physical Port....266
brocade-interface/gigabitethernet....267
Examples....268
Retrieving Gigabit Ethernet Port Statistics....268
Retrieving Information for a Single Gigabit Ethernet Port....269
Resetting the Port Stats for a Single Gigabit Ethernet Port....270
brocade-interface/portchannel....271
Examples....271
Displaying Port Channel Details....272
Configuring a Port Channel....273
brocade-license....274
Examples....275
Viewing the License Data....275
Viewing the Port License Assignments....276
Viewing the Brocade End User License Agreement....278
brocade-lldp....282
Examples....282
Viewing Global LLDP Information for a Switch....283
Configuring an LLDP Profile....283

Monitoring an LLDP Neighbor....284
Viewing LLDP Statistics....284
brocade-logging....286
Examples....288
Viewing the Current Audit Log Configuration....293
Enabling Audit Logging....294
Disabling Audit Logging....294
Configuring the Security Level....295
Configuring the Filter Class....295
Viewing the Current Syslog Configuration....295
Configuring a Non-secure Syslog Server....296
Configuring a Secure Syslog Server....297
Removing a Syslog Server....297
Viewing all External RASlog Messages....297
Viewing the Status of a Specific RASlog Message....299
Enabling Logging for a Specific RASlog Message....299
Disabling Logging for a Specific RASlog Message....299
Enabling Syslog for a Specific Internal RASlog Message....300
Disabling Syslog for a Specific RASlog Message....300
Configuring Settings for a Specific External RASlog Message....300
Configuring Settings for a Specific Internal RASlog Message....301
Viewing a List of the Current RASlog Modules....301
Enabling Logging for all Messages in a RASlog Module Group....302
Disabling Logging for all Messages in a RASlog Module Group....303
Viewing the Quiet Time Configuration....303
Enabling Quiet Time for a Specific Log Type....303
Disabling Quiet Time for a Specific Log Type....304
Configuring the Quiet Time....304
Configuring Quiet Time for a Specific Log Type with Start Time, End Time, and Days of Week Specified....305
Configuring the Syslog Facility....305
Configuring the Keepalive Period....306
Viewing the Support FTP Settings....306
Configuring Support FTP....306
Viewing Audit Logs....307
Viewing Error Logs....308
brocade-management-ip-interface....310
Examples....311
Viewing the LLDP Statistics for the Management Interface....311
Viewing the LLDP Neighbor Devices for the Management Interface....311
brocade-maps....313

Examples....318
Viewing the Switch Status Policy Report....318
Viewing System Resources....318
Viewing the Paused Configuration....319
Pausing MAPS Monitoring....320
Resuming MAPS Monitoring....320
Viewing MAPS Groups....321
Configuring a New MAPS Group....322
Modifying an Existing MAPS Group....322
Changing the MAPS Configuration....323
Viewing the MAPS Dashboard....323
Viewing MAPS Dashboard Settings....326
Clearing the MAPS Dashboard Data....326
Viewing MAPS Rules....327
Configuring a MAPS Rule....328
Manually Clearing Quiet Time....329
Manually Clearing Quarantined Ports....329
Configuring a MAPS Policy....329
Activating a MAPS Policy....330
Viewing the Credit Stall Dashboard....330
Viewing the Oversubscription Dashboard....331
Viewing the Dashboard History....332
Configuring an FPI Profile....333
Viewing the MAPS Device Login Data....334
Viewing the Backend Port History....334
Viewing the Gigabit Ethernet Ports History....335
brocade-media....337
Examples....339
Viewing the Temperature Alerts for a Specific Interface....339
brocade-name-server....341
Examples....341
Retrieving Name Server Information for a Switch....342
Retrieving Name Server information for a Port....343
brocade-security....345
brocade-security Examples....352
Viewing All IP Filter Rules....390
Configuring an IP Filter Policy....391
Cloning an IP Filter Policy....391
Activating an IP Filter Policy....392
Deleting an IP Filter Policy....392

Viewing all IP Filter Policies....392
Configuring an IP Filter Rule....393
Deleting an IP Filter Rule....394
Viewing Password Configurations for All Users on a Device....394
Configuring a Password Configuration for a New User....395
Deleting a Password Configuration for a User on a Device....395
Deleting All Password User Policies....396
Deleting All Password User Policies....396
Viewing the Password Policy....396
Viewing the Password Policy for a Specific User....397
Configuring the Password Policy....398
Viewing User Configurations....399
Configuring a New User/Editing an Existing User....400
Viewing the Authentication Configuration....400
Configuring the Authentication Configuration....401
Viewing the RADIUS Server Configuration....401
Configuring a RADIUS Server....402
Deleting a RADIUS Server....403
Viewing the TACACS+ Server Configuration....403
Configuring a TACACS+ Server....404
Deleting a TACACS+ Server....404
Viewing the LDAP Server Configuration....405
Configuring a LDAP Server....405
Deleting a LDAP Server....406
Viewing LDAP Role Mappings....406
Mapping LDAP Roles to Switch Roles....407
Deleting a LDAP Role Mapping....407
Viewing the Security Crypto Configuration Templates....407
Viewing the Security Crypto Configuration....410
Importing a SecCryptoCfg Template....411
Exporting a SecCryptoCfg Template....411
Activating a SecCryptoCfg Template....412
Verifying a SecCryptoCfg Template....412
Viewing the Allowed User and Rekey Interval Configuration....413
Configuring a User for Public Key and Rekey Interval Configuration....413
Generating a Host Key....414
Deleting a Host Key....414
Generating a Private Key....415
Deleting a Private Key....415
Viewing Public Keys....415

Deleting a Public Key for a User....416
Importing a Public Key from a Remote Host....416
Adding a Remote Host Key to a Known Host File....417
Exporting a Public Key to a Remote Host....417
Configuring a User Password....418
Generating a Third-Party Security Certificate Request on the Switch....419
Generating a Self-Signed Security Certificate on the Switch....419
Importing a Security Certificate....420
Exporting a Security Certificate....421
Deleting a Security Certificate....422
Viewing all Security Certificates on a Device....422
Viewing a List of Security Policies....427
Viewing the Defined DCC Policies....427
Viewing the Defined FCS Policy....428
Viewing the Defined SCC Policy....429
Viewing a List of the Active DCC Policy Members....429
Viewing a List of the Active FCS Policy Members....430
Viewing a List of the Active SCC Policy Members....431
Viewing the Policy Distribution Configuration....431
Editing the Policy Distribution Configuration....432
Viewing the Security Database Size....432
Viewing the Security Statistics....433
Viewing the DH-CHAP Shared Secret Key....434
Viewing the Authentication Configuration....434
Editing the Authentication Configuration....435
Viewing a List of the RBAC Permissions for the MOF Classes and CLI Commands....435
Viewing the RBAC Permissions....436
Viewing a List of Extension-related Security Certificates....438
brocade-snmp....442
Examples....443
Viewing SNMP System Settings....443
Configuring SNMP System Settings....443
Viewing MIB Configurations....444
Enabling an MIB....445
Disabling an MIB....445
Viewing the Configuration for Traps....446
Configuring a Trap....447
Viewing the SNMPv1 Account Settings....447
Configuring a SNMPv1 Account....448
Viewing SNMPv1 Trap Settings....448

Configuring a SNMPv1 Trap....450
Viewing the SNMPv3 Account Settings....450
Configuring a SNMPv3 Account....451
Viewing SNMPv3 Trap Settings....451
Configuring a SNMPv3 Trap....453
Viewing SNMP Access Control Settings....453
Configuring SNMP Access Control Settings....454
brocade-supportlink....456
Examples....456
Enabling Support Link....456
Configuring the Support Link....457
Displaying the Support Link Configuration....457
Displaying the Support Link History....458
brocade-time....460
Examples....461
Viewing the Time Zone....461
Configuring the Time Zone Offset....461
Configuring the Time Zone Name....462
Viewing the Clock Server Configuration....462
Configuring the Clock Server....462
brocade-traffic-optimizer....464
Examples....465
Viewing All Traffic Optimizer Counters for a Performance Group....465
Viewing All Flows for a Specific Traffic Optimizer Performance Group....465
brocade-usb....466
Examples....466
Viewing the Files and Folders on an Attached USB....466
Enabling an Attached USB....467
Disabling an Attached USB....467
brocade-zone....468
Examples....469
Retrieving the List of Zone Configurations....469
Retrieving the List of Member Zones....470
Retrieving a List of the Zone Members in a Specific Zone....471
Adding a Zone to the Defined Configuration....471
Deleting a Zone from the Defined Configuration....472
Creating an Alias....472
Viewing the Fabric Lock Data....473
Sample Use Cases....474
Logging On, Retrieving Switch Information, and Logging Out....474

Logging On, Retrieving Chassis Information, and Logging Out Using Session-Less REST API Operation.....476
Creating a Port Trunk Area Using JSON.....476
Disabling and Enabling a Port.....480
Creating an Up State Tunnel Using Multiple URIs.....481
Running a ClearLink Diagnostic Port Test.....484
Creating a MAPS Rule to Monitor CRC Errors on FC Ports.....486
Monitoring the Execution of a MAPS Rule.....488
Creating a User-Defined Group, Adding Ports to the Group, and Using the Group to Monitor a Rule.....489
Retrieving the Switch Status Policy Report.....490
Generating a CSR and Importing a Security Certificate.....491
Configuring SSH Public Key Authentication on a Switch for Incoming Connections.....493
Configuring SSH Public Key Authentication on a Switch for Outgoing Connections.....494
Creating a New Zone Using REST.....495
Adding Additional Zone Members to an Existing Zone.....496
Creating, Modifying, and Deleting a Zone Using REST.....497
Creating a New Zone in an Existing Configuration Using REST.....497
Modifying a Zone.....502
Aborting a Zone Transaction.....504
Deleting a Zone.....505
Concurrent Zoning Transactions on a Local Switch.....507
Commitment of Simultaneous Multiswitch Zone Transactions.....508
Timeout of the REST Zone Transaction Timer.....508
REST Running Config.....510
brocade-access-gateway/device-list.....510
brocade-access-gateway/f-port-list.....511
brocade-access-gateway/n-port-settings.....513
brocade-access-gateway/policy.....513
brocade-access-gateway/port-group.....515
brocade-application-server/application-server-device.....517
brocade-chassis/chassis.....518
brocade-chassis/credit-recovery.....522
brocade-chassis/ha-status.....524
brocade-chassis/management-ethernet-interface.....525
brocade-chassis/management-interface-configuration.....529
brocade-chassis/management-port-connection-statistics.....531
brocade-chassis/management-port-track-configuration.....533
brocade-chassis/sn-chassis.....533
brocade-chassis/version.....534
brocade-extension-ip-route/extension-ip-route.....535
brocade-extension-ipsec-policy/extension-ipsec-policy.....537

brocade-extension-tunnel/circuit-qos-statistics....539
brocade-extension-tunnel/extension-circuit....541
brocade-extension-tunnel/extension-circuit-statistics....547
brocade-extension-tunnel/extension-tunnel....548
brocade-extension-tunnel/extension-tunnel-statistics....558
brocade-extension-tunnel/wan-statistics-v1....560
brocade-extension/dp-hcl-status....562
brocade-extension/global-lan-statistics....563
brocade-extension/lan-flow-statistics....568
brocade-extension/traffic-control-list....572
brocade-fabric-traffic-controller/fabric-traffic-controller-ag-f-port....577
brocade-fabric-traffic-controller/fabric-traffic-controller-ag-n-port....579
brocade-fabric-traffic-controller/fabric-traffic-controller-device....580
brocade-fabric/access-gateway....584
brocade-fabric/fabric-switch....586
brocade-fdmi/hba....587
brocade-fdmi/port....591
brocade-fibrechannel-configuration/chassis-config-settings....596
brocade-fibrechannel-configuration/f-port-login-settings....597
brocade-fibrechannel-configuration/fabric....599
brocade-fibrechannel-configuration/port-configuration....600
brocade-fibrechannel-configuration/switch-configuration....601
brocade-fibrechannel-configuration/zone-configuration....604
brocade-fibrechannel-diagnostics/fibrechannel-diagnostics....604
brocade-fibrechannel-logical-switch/fibrechannel-logical-switch....610
brocade-fibrechannel-routing/edge-fabric-alias....612
brocade-fibrechannel-routing/fibrechannel-router....613
brocade-fibrechannel-routing/lsan-device....614
brocade-fibrechannel-routing/lsan-zone....616
brocade-fibrechannel-routing/proxy-config....617
brocade-fibrechannel-routing/router-statistics....618
brocade-fibrechannel-routing/routing-configuration....619
brocade-fibrechannel-routing/stale-translate-domain....622
brocade-fibrechannel-routing/translate-domain-config....623
brocade-fibrechannel-switch/fibrechannel-switch....624
brocade-fibrechannel-switch/switch-fabric-statistics....629
brocade-fibrechannel-switch/topology-domain....630
brocade-fibrechannel-switch/topology-error....631
brocade-fibrechannel-switch/topology-route....632
brocade-fibrechannel-trunk/performance....634

brocade-fibrechannel-trunk/trunk **635**
brocade-fibrechannel-trunk/trunk-area **637**
brocade-ficon/cup **638**
brocade-ficon/lirr **641**
brocade-ficon/logical-path **642**
brocade-ficon/rlir **644**
brocade-ficon/rnid **646**
brocade-ficon/switch-rnid **648**
brocade-firmware/firmware-config **650**
brocade-firmware/firmware-history **651**
brocade-fru/blade **652**
brocade-fru/fan **657**
brocade-fru/history-log **658**
brocade-fru/power-supply **660**
brocade-fru/sensor **662**
brocade-fru/wwn **663**
brocade-interface/extension-ip-interface **665**
brocade-interface/fibrechannel **667**
brocade-interface/fibrechannel-lag **687**
brocade-interface/fibrechannel-router-statistics **688**
brocade-interface/fibrechannel-statistics **689**
brocade-interface/gigabitethernet **695**
brocade-interface/gigabitethernet-statistics **698**
brocade-interface/logical-e-port **701**
brocade-interface/portchannel **702**
brocade-interface/portchannel-statistics **703**
brocade-license/end-user-license-agreement **705**
brocade-license/license **708**
brocade-license/ports-on-demand-license-info **710**
brocade-lldp/lldp-global **714**
brocade-lldp/lldp-neighbor **716**
brocade-lldp/lldp-profile **717**
brocade-lldp/lldp-statistics **718**
brocade-logging/audit **719**
brocade-logging/audit-log **720**
brocade-logging/error-log **722**
brocade-logging/log-quiet-control **724**
brocade-logging/log-setting **725**
brocade-logging/management-session-login-information **726**
brocade-logging/raslog **728**

brocade-logging/raslog-module....730
brocade-logging/supportftp....731
brocade-logging/syslog-server....732
brocade-management-ip-interface/management-interface-lldp-neighbor....733
brocade-management-ip-interface/management-interface-lldp-statistics....735
brocade-management-ip-interface/management-ip-interface....736
brocade-maps/backend-ports-history....739
brocade-maps/credit-stall-dashboard....739
brocade-maps/dashboard-history....741
brocade-maps/dashboard-misc....743
brocade-maps/dashboard-rule....745
brocade-maps/fpi-profile....747
brocade-maps/gigabit-ethernet-ports-history....748
brocade-maps/group....749
brocade-maps/maps-config....751
brocade-maps/maps-device-login....753
brocade-maps/maps-policy....754
brocade-maps/monitoring-system-matrix....757
brocade-maps/oversubscription-dashboard....761
brocade-maps/paused-cfg....762
brocade-maps/quarantined-devices....763
brocade-maps/rule....765
brocade-maps/switch-status-policy-report....767
brocade-maps/system-resources....769
brocade-media/media-rdp....770
brocade-name-server/fibrechannel-name-server....779
brocade-security/acl-policy....784
brocade-security/active-dcc-policy-member-list....785
brocade-security/active-fcs-policy-member-list....786
brocade-security/active-scc-policy-member-list....786
brocade-security/auth-spec....787
brocade-security/authentication-configuration....789
brocade-security/defined-dcc-policy-member-list....790
brocade-security/defined-fcs-policy-member-list....791
brocade-security/defined-scc-policy-member-list....792
brocade-security/dh-chap-authentication-secret....793
brocade-security/ipfilter-policy....794
brocade-security/ipfilter-rule....796
brocade-security/ldap-role-map....798
brocade-security/ldap-server....799

brocade-security/management-rbac-map ... 801
brocade-security/password ... 802
brocade-security/password-cfg ... 802
brocade-security/policy-distribution-config ... 806
brocade-security/radius-server ... 807
brocade-security/rbac-class ... 808
brocade-security/role-config ... 810
brocade-security/sec-crypto-cfg ... 814
brocade-security/sec-crypto-cfg-template ... 817
brocade-security/sec-crypto-cfg-template-action ... 820
brocade-security/security-certificate-action ... 821
brocade-snmp/access-control ... 822
brocade-snmp/mib-capability ... 823
brocade-snmp/system ... 824
brocade-snmp/trap-capability ... 826
brocade-snmp/v1-account ... 827
brocade-snmp/v1-trap ... 829
brocade-snmp/v3-account ... 830
brocade-snmp/v3-trap ... 832
brocade-supportlink/supportlink-history ... 834
brocade-supportlink/supportlink-profile ... 835
brocade-time/clock-server ... 837
brocade-time/ntp-clock-server ... 838
brocade-time/ntp-clock-server-key ... 839
brocade-time/time-zone ... 841
brocade-traffic-optimizer/partial-performance-group ... 842
brocade-traffic-optimizer/performance-group ... 843
brocade-traffic-optimizer/performance-group-flows ... 846
brocade-traffic-optimizer/performance-group-profile ... 846
brocade-usb/usb-file ... 848
brocade-zone/defined-configuration ... 849
brocade-zone/effective-configuration ... 852

REST RPC Operations ... 857

abort ... 857
configdownload ... 857
configupload ... 859
date ... 860
device-login-rebalance ... 860
device-management ... 861
extension ... 861

fibrechannel-fabric 863
factory-reset 863
firmwaredownload 865
license 866
lldp 868
login 869
management-ethernet-interface 870
maps-policy-restore 870
maps-rule-restore 871
ntp-clock-server 871
pcie-health-test 872
port 872
port-decommission 873
portchannel 874
reboot 875
restart 876
sdd-quarantine 877
security-acl-policy 877
security-authentication-configuration 878
security-authentication-secret 879
security-certificate 880
security-fabric-wide-policy-distribute 881
security-policy-chassis-distribute 881
security-policy-distribute 882
security-reset-violation-statistics 883
security-role-clone 884
show-status 884
slot-configuration 886
supportlink 886
traffic-optimizer 887
usb-delete-file 888
fibrechannel-zone 888
Definitions 890
aaa-authspec-type 890
aaa-encryption-algorithm-type 890
aaa-protocols-type 890
aaa-timeout-type 890
access-permission 890
aaa-authspec-type 891
authentication-data 891

ag-mode-type ... **891**
air-flow-direction-type ... **891**
alarm-type ... **891**
alert-type ... **892**
area-address-type ... **892**
arl-algo-type ... **892**
as-number ... **893**
ascii-text-type ... **893**
authentication-hash-type ... **893**
authentication-initiation-parameter-type ... **894**
authentication-initiation-type ... **894**
authentication-operation-type ... **894**
authentication-policy-mode-type ... **894**
authentication-protocol-type ... **895**
authentication-secret-action-type ... **895**
authentication-secret-type ... **895**
base64-string-type ... **895**
blade-state-type ... **896**
blade-type-type ... **896**
boot-up-self-test-status-type ... **896**
brocade-empty-string-type ... **897**
brocade-hex-string-type ... **897**
brocade-telemetry-profile-operational-status-type ... **897**
brocade-telemetry-profile-state-type ... **897**
brocade-telemetry-profile-status-message-type ... **898**
brocade-telemetry-protocol-type ... **898**
brocade-telemetry-server-status-type ... **898**
brocade-telemetry-topic-type ... **899**
cable-descriptor-type ... **899**
capable-speed-type ... **899**
certificate-application-type ... **900**
certificate-entity-type ... **900**
certificate-hexdump-type ... **900**
cfg-action-type ... **901**
chassis-distribute-domain-type ... **901**
chassis-distribute-policy-type ... **901**
chassis-user-friendly-name-type ... **901**
circuit-admin-status-type ... **902**
class-of-service-type ... **902**
class-type ... **902**

clear-feature-type 903
comments-type 903
compression-protocol-type 904
compression-tunnel-type 904
congestion-detection-signal-cycles-type 904
congestion-detection-signal-scale-type 905
congestion-detection-signal-type 905
congestion-state-type 905
connection-state-type 906
control-qos-type 907
counter32 908
counter64 909
cp-name-type 909
credit-stall-state-type 910
crypto-version-type 910
current-type 910
data-type-type 910
date-and-time 911
date-code-type 912
dcc-policy-member-type 912
default-control-type 912
default-string-type 912
deskew-type 912
dh-group-type 913
diagnostics-operation-action-type 913
discovered-mtu-type 913
distribute-policy-type 914
domain-id-type 914
domain-index-type 914
domain-name 915
dotted-quad 916
dp-actstate-type 916
dp-dp-communication-type 917
dp-hcl-stage-type 918
dp-id-type 918
dp-state-type 919
dscp 920
dynamic-load-sharing-level-type 920
els-descriptor-type 920
error-stats-type 921

eula-action-type **922**
event-class-type **922**
extension-certificate-entity-type **922**
extension-ipsec-profile-name **923**
extension-lan-interface-type **923**
extension-name-type **923**
extension-qos-type **923**
fabric-id-type **923**
fabric-incident-type **923**
fabric-wide-consistency-policy-type **924**
fc-lag-link-status-type **924**
fc-qos-type **924**
fc-speed-type **925**
fc4-features-type **925**
fc4-type-type **925**
fcid-hex-string-type **926**
fcid-type **926**
fcr-device-state **926**
fec-type **926**
fibre-channel-router-owner **927**
fibrechannel-trunk-type **927**
firmware-version-type **927**
fru-date-type **928**
fru-state-type **928**
fru-type **928**
gauge32 ..... **929**
gauge64 ..... **930**
ge-interface-type **930**
ge-protocol-type **930**
gen-certificate-entity-type **931**
group-name **931**
hash-algorithm-type **931**
hcl-status-type **932**
hex-string **932**
home-virtual-fabric-type **932**
host **933**
host-type **933**
interface-type **933**
ip-address **933**
ip-address-no-zone **933**

ip-extension-interface-type 934
ip-gateway-type 934
ip-prefix 934
ip-prefix-length 934
ip-qos-type 935
ip-version 935
ipfilter-action-type 935
ipfilter-ip-version-type 936
ipfilter-name-type 936
ipfilter-operation-action-type 936
ipfilter-permission-type 936
ipfilter-protocol-type 936
ipfilter-traffic-type 936
ipv4-address 937
ipv4-address-no-zone 937
ipv4-prefix 937
ipv4-prefix-length 938
ipv6-address 938
ipv6-address-no-zone 938
ipv6-flow-label 939
ipv6-prefix 939
ipv6-prefix-length 939
keep-feature-type 940
keysize-type 940
known-host-type 940
l4-common-protocol 941
l4-protocol-number 941
l4-protocol-type 941
lag-port-timeout 941
lan-interface-type 941
ldap-tls-mode-type 942
link-fault-descriptor-group 942
lldp-mandatory-tlv-name-type 942
lldp-neighbor-parameters-group 942
lldp-optional-tlv-name-type 943
lldp-statistics-parameters-group 943
logical-path-operational-states 944
long-distance-type 944
los-tov-mode-type 944
mac-address 945

management-port-connection-group 945
manufacturer-type 946
maps-dashboard-category-type 947
maps-data-unit 948
maps-db-date-type 949
maps-event-severity-type 949
maps-generic-action-type 950
maps-group-feature-type 951
maps-group-name-type 951
maps-group-type-type 952
maps-logical-operator-type 953
maps-monitoring-system-type 953
maps-policy-name-type 953
maps-port-monitoring-state 953
maps-quiet-time-unit-type 954
maps-rule-name-type 954
maps-rule-type 954
maps-time-base-type 954
media-alerts-group 954
media-optical-product-data-group 955
media-power-type 955
message-id-type 955
mibs-name 956
module-id-type 956
monitoring-type-type 956
month-type 956
ms-acl-access-type 957
notification-capabilities-type 957
ntp-clock-server-group 957
object-identifier-128 958
object-identifier 959
octet-member-index-type 959
octet-speed-combo-type 960
operational-state-type 960
operational-status-type 960
operational-type 961
optical-type 962
oversubscription-state-type 962
part-number-type 962
password-cfg-operation-type 962

percentage-type 962
performance-group-id-type 963
phys-address 963
physical-interface-type 963
pki-key-pair 963
port-address-binding-mode 964
port-decommission-operation-type 964
port-group-state-type 964
port-number-type 964
port-number 965
port-operation-type 965
port-scn-type 966
port-type-string-type 967
port-type-type 968
port-type 968
portchannel-interface-name-type 968
portchannel-speed-type 969
portchannel-type 969
portchannel_platform 969
power-detail-type 969
power-state-type 969
preshared-key 970
privacy-protocol-type 970
product-name-type 970
protocol-speed-type 971
protocol 972
rbac-class-permission-type 972
rbac-class-type 972
rbac-context-type 972
rbac-permission-type 972
rbac-role-type 973
reboot-types 973
reliable-type 973
result-type 974
role-name-type 974
sec-crypto-cfg-actions-type 974
sec-crypto-cfg-default-template-name-type 975
sec-crypto-cfg-file-transfer-protocol-type 975
sec-crypto-cfg-template-name-type 975
sec-crypto-cfg-tls-cipher-type 975

seccertmgmt-action-type **975**
seccertmgmt-hash-type **976**
seccertmgmt-operation-type **976**
seccertmgmt-protocol-type **976**
seccrypto-import-export-param-group **976**
security-compliance-action-type **978**
security-info-group **978**
security-input-type **979**
security-level-string-type **979**
security-level **979**
security-policy-action-type **980**
security-policy-default-name-type **980**
security-policy-distribute-permission-type **980**
security-policy-name-type **980**
security-policy-state-type **981**
security-violation-statistics-type **981**
segmentation-reason-type **982**
sensor-category **983**
serial-number-type **983**
server-parameters **983**
severity-level-type **984**
severity-level **984**
show-status-group **984**
signal-descriptor-type **985**
slot-configuration-action-type **985**
slot-port-name-type **986**
speed-type **986**
sshutil-algorithm-type **986**
sshutil-hash-type **986**
sshutil-key-type **986**
sshutil-operation-type **987**
ssp-state-type **987**
state-type **987**
stream-type **987**
switch-addressing-mode-type **988**
tcl-action-type **988**
tcp-operation-mode **988**
temp-alarm-type **988**
temperature-type **989**
threshold-value-type **989**

time-24hr-extended-type 989
time-24hr-type 989
time-detail-type-in-hours 989
time-detail-type 989
time-generated-type 990
timestamp 990
timeticks 991
tls-protocol-type 991
topology-report-error-type 991
transmission-types 991
traps-name 992
ts-authspec-type 993
ts-key-type 993
ts-ntp-type 993
ts-operation-type 993
ts-timezone-type 994
tunnel-ha-type 995
tunnel-load-level 996
uri 996
usb-filename-type 997
usb-path-type 997
user-config-access-hours-type 997
user-config-access-minutes-type 997
user-config-login-user-name-type 997
user-config-role-type 997
user-config-user-name-type 997
user-password-type 998
user-port-number-type 998
uuid 998
validation-mode-type 998
ve-interface-type 998
vendor-name-type 998
vendor-revision-type 999
virtual-channel-type 999
virtual-fabric-action-type 999
virtual-fabric-configuration-group 999
virtual-fabric-role-id-type 1000
voltage-type 1000
wwn-type 1000
xpath1.0 1000

yang-identifier **1001**
zero-based-counter32 ..... **1001**
zero-based-counter64 ..... **1002**
zone-member-type **1002**
zone-operation-action-type **1003**
zone-type-string-type **1004**
zone-type-type **1004**
zoning-name-type **1004**
References **1005**
REST API Description **1005**
Resources 1005
Base Resources 1006
Resources and the YANG Model 1006
Relationship between the YANG and Resource Data Models 1006
Uniform Resource Identifiers 1007
URI Structure 1007
XML Resource Representation **1009**
JSON Resource Representation **1009**
Error Reporting **1011**
Error Returned Due to an Invalid POST Request 1013
Existing IP Address Error 1014
Unsupported Platform Error 1015
Login Errors 1015
Deprecated and Obsolete Resources in Fabric OS 9.2.0 ..... **1017**
Revision History **1021**
Documentation Legal Notice **1022**

Introduction

This document describes the REST API (a programmable Web service interface) for Brocade® Fabric OS® (FOS). The REST API can manage Brocade SAN switches across a fabric.

Supported Hardware and Software

The following hardware platforms are supported by Brocade Fabric OS 9.2.x.

Although many different software and hardware configurations are tested and supported by Broadcom Inc. for Fabric OS 9.2.x, documenting all possible configurations and scenarios is beyond the scope of this document.

Brocade Gen 7 (64G) Fixed-Port Switches

- Brocade G720 Switch
- Brocade G730 Switch
- Brocade 7850 Extension Switch

Brocade Gen 7 (64G) Directors

- Brocade X7-4 Director
- Brocade X7-8 Director

Brocade Gen 6 (32G) Fixed-Port Switches

- Brocade G610 Switch
- Brocade G620 Switch
- Brocade G630 Switch
- Brocade 7810 Extension Switch

Brocade Gen 6 (32G) Directors

- Brocade X6-4 Director
- Brocade X6-8 Director

Brocade REST API Courses

Brocade provides online training for the FOS REST API, including implementation, sending requests, and automation.

1. Go to the Brocade course catalog (https://www.broadcom.com/support/education/brocade/courses).
2. Click **Technologies**.
3. Click **Brocade Automation** playlist.
4. Select the course you want to run.

REST Terminology

The following terms are used in this manual to describe REST and REST functionality contained in the Brocade FOS REST API.

base URI	A base URI is specific to the Fabric OS server. All URIs accessing the same server use the same base URI. For example, in the request “POST http://10.10.10.10/rest/login”, “http://10.10.10.10/rest/” is the base URI.
device	Any host or target device with a distinct WWN. Devices may be physical or virtual.
D_Port	A port configured as a diagnostic port on a switch or connected fabric switch, used to run diagnostic tests between ports to test the link.
E_Port	An interswitch link (ISL) port. A switch port that connects switches together to form a fabric.
F_Port	A fabric port. A switch port that connects a host, host bus adapter (HBA), or storage device to a SAN.
fabric	A fabric consists of interconnected nodes that look like a single logical unit when viewed collectively. This refers to a consolidated high-performance network system consisting of coupled storage devices, networking devices, and parallel processing high bandwidth interconnects such as 8Gb/s, 10Gb/s, 16Gb/s, or 32Gb/s Fibre Channel ports.
FCID	A Fibre Channel ID (FCID) is a 24-bit (3 byte) field used to route frames through a Fibre Channel (FC) network. The FOS REST API shows this as a decimal value, such as “14776283”.
FCIP	Fiber Channel over IP. This functionality allows connectivity between two remote fabrics that are separated by an IP network. FCIP is different from “IP over FC”. Refer to RFC 3821.
HTTP	HyperText Transfer Protocol. An application protocol for distributed, collaborative, and hypermedia information systems. Refer to RFC 7230.
Internet Protocol (IP)	The principal communications protocol in the Internet Protocol suite. It conveys packets from a source host to a destination host based on the IP addresses contained in the packet headers. Refer to RFC 791.
IPsec	Internet Protocol security. A network protocol suite that authenticates and encrypts the packets of data sent over a network, protecting data flows between a pair of hosts (host-to-host), between a pair of security gateways (network-to-network), or between a security gateway and a host (network-to-host). Refer to RFC 4301, RFC 4303, and RFC 4309.
JSON	JavaScript Object Notation. JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser–server communication, including as a replacement for XML in some AJAX-style systems. Refer to RFC 8259 and ECMA-404.
N_Port	A node port. An N_Port presents a host or storage device to the fabric.
NETCONF	NETwork CONFiguration protocol. A network management protocol developed and standardized by the IETF. Refer to RFC 6241.
NPIV	N_Port ID Virtualization. NPIV is a Fibre Channel facility that allows multiple F_Port IDs to share a single physical N_Port. Multiple F_Ports can be mapped to a single N_Port. This allows multiple Fibre Channel initiators to occupy a single physical port, easing hardware requirements in storage area network design, especially for virtual SANs.
request URI	A request URI is the URI used to perform a REST HTTP request such as GET, POST, PUT, DELETE, HEAD or OPTIONS.
REST	REpresentational State Transfer is a way of providing interoperability between computer systems in a network.
RESTCONF	REST CONFiguration protocol. A RESTful protocol used to access data defined using the YANG language. Refer to RFC 8040.
SNMP	Simple Network Management Protocol. SNMP is a set of protocols for managing complex networks. SNMP protocols are application layer protocols. Using SNMP, devices within a network send messages, called protocol data units (PDUs), to different parts of a network. Refer to RFC 1157, RFC 3411, RFC 3412, RFC 3414, and RFC 3415.
TCP	Transmission Control Protocol. One of the main protocols of the Internet Protocol (IP) suite. TCP provides reliable, ordered, and error-checked delivery of a stream of data in octets between applications running on hosts communicating by an IP network. Refer to RFC 1122 and RFC 7323.
VE_Port	Virtual E_Port. Software mechanism for creating a logical E_Port connection to allow the Fibre Channel fabric protocols to communicate over this virtual interface. Typically used for FCIP products.

YANG	Yet Another Next Generation. A data modeling language for the definition of data sent over the NETCONF network configuration protocol. Refer to RFC 7950.
XML	eXtensible Markup Language. A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable through use of tags that can be created and defined by users. Refer to the W3C XML standard.

NOTE
Definitions are based on information extracted from either the specific standard or the Wikipedia entry for that term.

For definitions of additional SAN-specific terms, visit the Storage Networking Industry Association online dictionary:

http://www.snia.org/education/dictionary

Terms Used in YANG Files

The following list provides definitions and explanations for terms commonly used when discussing YANG files. Refer to RFC 6020 for complete details on YANG language features.

augmentation	The module hierarchy can be augmented, allowing one module to add data nodes to the hierarchy defined in another module (without changing the augmented module). Augmentation can be conditional, using when statements, with new nodes appearing only if certain conditions are met.
case node	YANG allows the data model to segregate incompatible nodes into distinct choices using “choice” and “case” statements. A “choice” statement contains a set of “case” statements that define sets of schema nodes that cannot appear together. Each “case” may contain multiple nodes, but each node may appear in only one “case” under a “choice”. When an element from one case is created, all elements from all other cases are implicitly deleted. The device handles the enforcement of the constraint, preventing incompatibilities from existing in the configuration. The choice and case nodes appear only in the schema tree, not in the data tree or in NETCONF messages. The additional levels of hierarchy are not needed beyond the conceptual schema. Refer to section 4.2.7 of RFC 6020.
choice node	This defines a set of alternatives, only one of which may exist at any one time. A choice consists of a number of branches, defined with case substatements. The nodes from one of the choice's branches (limited to one) exist in the data tree, and the choice node itself does not exist in the data tree.
container	A container node has at most one instance in a module.
data model	A data model describes how data is represented and accessed. In YANG, data models are represented by definition hierarchies called schema trees. Instances of schema trees are called data trees and are encoded in XML.
data node	Data nodes can represent either configuration data or state data. The config statement specifies whether the definition it occurs in represents configuration data (config=true) or status data (config=false). If config is not specified, the default is the same as the parent schema node’s config value. If the top node does not specify a config statement, the default is config=true.
grouping	A grouping defines a reusable collection of nodes.
if	An “if” statement in a `.yang` file indicates that the leaf or container has conditional support. Refer to section 7.18.2 of RFC 6020.
key	A key provides a unique identifier for an entry in a list. A key may be composed of a single leaf, such as “name”, or by a combination of leaves such as “name” plus “address”.
leaf	A leaf node has at most one instance in a container or list.
leaf-list	A leaf-list node may have multiple instances.
leafref	A leafref is used to reference a particular leaf instance in the data tree, as specified by a path. This path is specified using the XML Path Language (XPath), in a notation that is similar to the syntax for directory paths in UNIX/Linux.

list	A list defines a sequence of list entries. Each entry is like a structure or a record instance, and the individual entry is uniquely identified by the values of its key leafs. A list can define multiple key leafs and may contain any number of child nodes of any type (including leafs, lists, containers).
mandatory	A mandatory statement indicates that a node must exist in the data tree. This applies only to POST requests.
module	YANG organizes data models into modules and submodules. A module can import data from other modules, and include data from submodules. Besides schema definitions, a module contains header statements (yang-version, namespace, prefix), linkage statements (import and include), meta-information (organization, contact), and a revision history.
presence	A presence statement provides a description of what the model designer intends the node's presence to signify.
schema tree	A schema tree defines the data structure for validation, documentation, and interaction control.
when	A "when" statement in a `.yang` file makes the node's parent data definition statement conditional. The node defined by the parent data definition statement is only valid when the condition specified by the "when" statement is satisfied. Refer to section 7.19.5 of RFC 6020.
union	A data type that is composed of multiple data types.

Standards and Related References

The following list identifies the IETF RFCs, ECMA, and W3C standards that apply to the FOS REST API:

- *User Datagram Protocol* (RFC 768)
- *Internet Protocol* (RFC 791)
- *Transmission Control Protocol* (RFC 793)
- *Requirements for Internet Hosts -- Communication Layers* (RFC 1122)
- *Requirements for Internet Hosts -- Application and Support* (RFC 1123)
- *A Simple Network Management Protocol (SNMP)* (RFC 1157)
- *Management Information Base for Network Management of TCP/IP-based internets: MIB-II* (RFC 1213)
- *Guidelines for creation, selection, and registration of an Autonomous System (AS)* (RFC 1930)
- *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms* (RFC 2001)
- *Internet Protocol, Version 6 (IPv6) Specification* (RFC 2460)
- *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers* (RFC 2474)
- *Structure of Management Information Version 2 (SMIv2)* (RFC 2578)
- *Textual Conventions for SMIv2* (RFC 2579)
- *IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers* (RFC 2780)
- *Textual Conventions for Additional High Capacity Data Types* (RFC 2856)
- *Management Information Base for the Differentiated Services Architecture* (RFC 3289)
- *Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations* (RFC 3305)
- *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks* (RFC 3411)
- *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* (RFC 3412)
- *Simple Network Management Protocol (SNMP) Applications* (RFC 3413)
- *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)* (RFC 3414)
- *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)* (RFC 3415)
- *Internet Protocol Version 6 (IPv6) Addressing Architecture* (RFC 3513)
- *Textual Conventions for IPv6 Flow Label (RFC 3595)* (RFC 3595)
- *Fibre Channel Over TCP/IP (FCIP)* (RFC 3821)
- *Uniform Resource Identifier (URI): Generic Syntax* (RFC 3986)
- *Textual Conventions for Internet Network Addresses* (RFC 4001)
- *IPv6 Scoped Address Architecture* (RFC 4007)
- *A Universally Unique IDentifier (UUID) URN Namespace* (RFC 4122)
- *A Border Gateway Protocol 4 (BGP-4)* (RFC 4271)
- *IP Version 6 Addressing Architecture* (RFC 4291)
- *Security Architecture for the Internet Protocol* (RFC 4301)
- *IP Encapsulating Security Payload (ESP)* (RFC 4303)
- *Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)* (RFC 4309)

- *Datagram Congestion Control Protocol (DCCP)* (RFC 4340)
- *Stream Control Transmission Protocol* (RFC 4960)
- *MIB Textual Conventions for Uniform Resource Identifiers (URIs)* (RFC 5017)
- *A Recommendation for IPv6 Address Text Representation* (RFC 5952)
- *YANG – A Data Modeling Language for the Network Configuration Protocol* (RFC 6020)
- *Guidelines for Authors and Reviewers of YANG Data Model Documents* (RFC 6087)
- *Network Configuration Protocol (NETCONF)* (RFC 6241)
- *BGP Support for Four-Octet Autonomous System (AS) Number Space* (RFC 6793)
- *Common YANG Data Types* (RFC 6991)
- *A YANG Data Model for Interface Management* (RFC 7223)
- *IANA Interface Type YANG Module* (RFC 7224)
- *A YANG Data Model for IP Management* (RFC 7227)
- *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing* (RFC 7230)
- *HTTP/1.1 Semantics and Content* (RFC 7231)
- *YANG System Management* (RFC 7317)
- *TCP Extensions for High Performance* (RFC 7323)
- *YANG Module Library* (RFC 7895)
- *The YANG 1.1 Data Modeling Language* (RFC 7950)
- *RESTCONF Protocol* (RFC 8040)
- *YANG Patch Media Type* (RFC 8072)
- *The JavaScript Object Notation (JSON) Data Interchange Format* (RFC 8259)
- *The JSON Data Interchange Syntax* (ECMA-404)
- *Uniform Resource Identifier (URI): Generic Syntax* (STD 66)
- *Extensible Markup Language (XML) 1.0* (W3C XML standard)

Contacting Technical Support for Your Brocade® Product

If you purchased Brocade® product support from a Broadcom® OEM or solution provider, contact your OEM or solution provider for all your product support needs.

- OEM and solution providers are trained and certified by Broadcom to support Brocade products.
- Broadcom provides backline support for issues that cannot be resolved by the OEM or solution provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information on this option, contact Broadcom or your OEM.
- For questions regarding service levels and response times, contact your OEM or solution provider.

If you purchased Brocade product support directly from Broadcom, use one of the following methods to contact the Technical Assistance Center 24x7. For product support information and the latest information on contacting the Technical Assistance Center, go to www.broadcom.com/support/fibre-channel-networking/contact-brocade-support.

Online	Telephone
For nonurgent issues, the preferred method is to log on to the Support portal at support.broadcom.com. (You must initially register to gain access to the Support portal.) Once registered, log on and then select **Brocade Products**. You can now navigate to the following sites: • **Case Management** • **Software Downloads** • **Licensing** • **SAN Reports** • **Brocade Support Link** • **Training & Education**	For Severity 1 (critical) issues, call Brocade Fibre Channel Networking Global Support at one of the phone numbers listed at www.broadcom.com/support/fibre-channel-networking/contact-brocade-support.

Document Feedback

Quality is our first concern. We have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission or if you think that a topic needs further development, we want to hear from you. Send your feedback to documentation.pdl@broadcom.com. Provide the publication title, publication number, topic heading, page number, and as much detail as possible.

Fabric OS REST API Overview

Fabric OS 8.2.0 and later support an application programming interface (API) for managing Brocade storage area network (SAN) switches. The FOS REST API is not supported for releases of Fabric OS prior to 8.2.0.

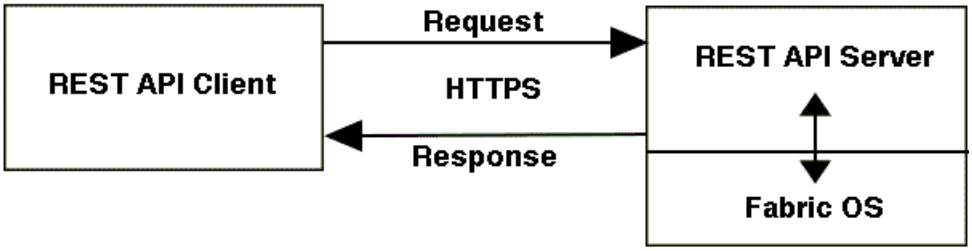
This API is referred to as the FOS REST API and is enabled by default in Fabric OS 8.2.0 and later. It follows the RESTCONF protocol defined in IETF RFC 8040 and defines data using the YANG 1.1 (Yet Another Next Generation) data modeling language as defined in IETF RFC 7950.

NOTE
Because the FOS REST API permits HTTP communications, it is not fully compliant with RFC 8040.

FOS REST API Described

The FOS REST API is a programmable web service interface for Brocade Fabric OS that can manage Brocade SAN switches across a fabric. This API uses standard HTTP methods to perform Create, Read, Update, and Delete (CRUD) operations on the fabric configuration data; and it provides an interface for provisioning, status, and validation operations using the YANG data model described in the YANG 1.1 RFC, but not the datastore managed with NETCONF. An Apache web server that is embedded in Fabric OS is used to serve the API.

Figure 1: FOS REST API Architecture



FOS REST API Session-Less Operation Described

FOS REST API session-less operation allows you to provide authentication credentials directly for each GET request. Essentially, FOS REST API session-less operation completes the login, GET operation, and logout as one complete request. You can only use basic authentication formats for REST API session-less operation, which includes plain text or Base64-encoded text.

NOTE
Fabric OS 8.2.2 or later support REST API session-less operation for GET requests only.

FOS REST API session-less operation requests do not count toward the maximum permitted number of REST API sessions. However, a Fabric OS switch must have at least one FOS REST API session available to execute session-less requests. The default limit for FOS REST API sessions is three sessions. A session-less REST API request opens an implicit active session that is closed at the end of the request.

FOS REST API session-less operation has the same hardware support and functionality requirements (unless otherwise noted) as FOS REST API. For more information about using FOS REST API session-less operation, see Using Brocade FOS REST API Session-Less Operation.

Software Support

The FOS REST API allows a user, application, or script to control certain aspects of a Brocade Gen 6 storage area network (SAN) device running Fabric OS 8.2.0 and later or a Gen 7 SAN device running Fabric OS 9.0.0 and later. Fabric OS 8.2.0 and later include a set of CLI commands that control the API functionality, including REST session management.

Hardware Support

The FOS REST API supports all Brocade Gen 6 Fibre Channel devices that are supported by Fabric OS 8.2.0 and later and Gen 7 devices that are supported by Fabric OS 9.0.0 and later. For a complete list of these platforms, refer to Supported Hardware and Software. The FOS REST API is not supported on the Brocade Analytics Monitoring Platform or the Brocade FX8-24 blade. The Brocade Extension platforms have some additional restrictions; these are noted in FOS REST API and Brocade Extension.

Functionality

The FOS REST API supports the following set of REST operations used for switch interaction: GET, POST, PATCH, DELETE, HEAD, and OPTIONS. REST sessions can be established through all supported management IP addresses. Attempting to use an unsupported method will return the HTTP status code 405, “Method Not Allowed”. No RASLog entry is made for this error.

The modules in Fabric OS 8.2.0 include support for fabric and switch discovery, zoning operations, port configuration, performing port diagnostics (D_Port operations), extension configuration and retrieving interface statistics.
Fabric OS 8.2.0a added support for access gateway, FDMI, Name Server, and logical switch. Fabric OS 8.2.1 added support for chassis, Fibre Channel configuration, FRUs, logging, media, operations (supportSave and show-status) trunking, time, security, and MAPS.

The modules in Fabric OS 9.0.x added support for Fabric Traffic Controller, Fibre Channel routing, FICON, SupportLink, Extension, and port channel, as well as additional Operations support (device management, extension, fabric, license, LLDP, PCIE health, SupportLink, and zoning).

The FOS REST API can also do the following:

- Restart Fabric OS Extension Internet Key Exchange (IKE) sessions.
- Reset interface statistics.
- Perform parallel REST virtual fabric operations across multiple REST sessions with multiple virtual fabrics.
- Operate in conjunction with Brocade SANnav™ Management Portal and Web Tools activity.

NOTE
The REST API for Brocade SANnav Management Portal is not the same as the FOS REST API, although there may be some overlaps. For REST operations using Brocade SANnav Management Portal, refer to the *Brocade® SANnav™ Management Portal REST API and Northbound Streaming Reference Manual*.

FOS REST API URI Restrictions

The following restrictions apply to FOS REST API URI requests.

- A FOS REST API request URI cannot be longer than 255 characters.
- A FOS REST API request URI cannot have more than 20 resource segments. A resource segment is the content between two slashes, and the count starts after the switch address.
  Examples: The request `POST http://10.10.10.10/rest/running/brocade-extension-ip-route/extension-ip-route` has 4 segments. The request `GET http://10.10.10.10/rest/running/brocade-zone/zoning/defined-configuration` has 5 segments.

The FOS REST API will reject requests and return a descriptive error for any of the following:

- Any REST operations that use unsupported HTTP methods (COPY, LINK, UNLINK, PUT, PURGE, LOCK, UNLOCK, PROPFIND, VIEW).
- Any URIs that do not meet length or part count requirements.
- Any URIs that do not contain all required keys at the same container level. However, the FOS REST API will accept URIs having multiple keys at the same container level in different orders.

For additional information on URI request structures, see URI Structure.

Security and the FOS REST API

FOS REST API function calls are permitted or denied based on user privilege configurations determined by the role-based access control (RBAC) functionality in Fabric OS. Only accounts with the following RBAC role permissions can log in using REST: admin, user, switchadmin, operator, zoneadmin, fabricadmin, basicswitchadmin, or securityadmin. There is no support for the following default switch roles: root and maintenance.

The following restrictions and constraints apply to using REST operations with Fabric OS:

- FOS REST API operations have the following restrictions:
  - REST operations are not permitted after the session timeout period expires. The default REST session timeout is 2 hours. This is updated if there is any change in the global http session timeout in configurechassis, webtools attributes command.
  - REST operations cannot be executed on a standby CP.
- FOS REST API logins have the following restrictions:
  - While the HTTP protocol is supported, Brocade strongly recommends using the HTTPS protocol for greater communication security before making REST calls.
  - HTTPS logins require a valid switch certificate. The login will fail if the switch certificate is not valid. A valid client certificate is not required, but is supported. Refer to the *Brocade Fabric OS Administration Guide* for instructions on installing certificates.
  - Neither HTTP and HTTPS logins are subject to time-of-day limits on account access.
  - A FOS REST API logout operation invalidates the session authorization key. After a REST logout, subsequent REST commands using the invalidated authorization key are not permitted.
  - FOS REST API logins are not persistent across switch reboots or HA failovers.

Only FOS REST API configuration activity is recorded as Fabric OS audit logs and RASLogs, these can be viewed using a GET request on /brocade-logging/audit-log or using the `auditdump -show` command. Read activity is not audited.

FOS REST API sessions are maintained if an Ethernet management interface cable is disconnected and reconnected.

Zoning Operations

The FOS REST API supports discovery and configuration of zone aliases, standard zones (both WWN and domain/port), QoS zones, LSAN zones, and peer zones.

You can use FOS REST API operations to perform the following zone operations:

- Zone creation
- Zone deletion
- Zone alias creation
- Zone alias deletion
- Zone configuration actions:
  - Zone configuration creation
  - Zone configuration deletion
  - Zone configuration enablement
  - Zone configuration disablement
  - Zone configuration addition
  - Zone configuration removal
  - Zone object member addition
  - Zone object member removal
  - Zone alias object member addition
  - Zone alias object member removal
- Zone database clear
- Zone database saving
- Zone transaction abort
- Zone database contents retrieval
- Default Zone mode modification to “NoAccess” or “AllAccess”

Zoning Restrictions

The following restrictions apply to zoning actions using the FOS REST API:

- Only zone discovery is supported for boot LUN zones, redirect and multi-service frame redirect (MSFR) zones, and traffic isolation (TI) zones.
- The maximum zone database transaction size that FOS API REST operations can handle is 4 MB.
- REST operations fail on zone operations using an incorrect zone database checksum.
  All REST clients are expected to store and provide the MD5 (128-bit) ZoneDB checksum in their zoning commit operations. If the provided checksum does not match the checksum stored in zoning, the commit operation will not be allowed. The intent of this requirement is to avoid management interfaces operating off of a zoneDB cache to push in stale data and thus potentially overwrite zone updates that were made after the zoneDB copy was synced.

Zoning Transaction Timer Restrictions

FOS API REST zoning transactions are assumed to be short-lived. However, there is no way to enforce this and clients can potentially open a zone transaction and leave it open indefinitely. This would lock out other zoning users from making changes on the local switch.

To prevent an indefinite lockout condition, zoning maintains a FOS API REST zone transaction timer. The timer is started the first time a FOS API REST zone transaction is opened and on each successive zoning operation, the timer is restarted. The timer is tied to the specific REST session ID. A client cannot start a zoning operation in one session, log out, login again under a different session ID and continue the zone transaction from the prior session. The switch treats the new login session as a different client and the previous dangling zone transaction is cancelled. For the pending

FOS API REST zoning operations, the zone fabric lock time out can be configurable and the default time out value is 5 minutes. After the default time out, a zoning transaction can be recommenced by the original REST client provided that neither of the following has happened:

- No other zone user on the local switch has tried to start another zone transaction.
- The transaction has not been aborted via a zone merge or remote zone commit operation.

NOTE
The zone fabric lock timeout value is configurable and the default time out value is 5 minutes.

RESTCONF transactions follow legacy zone transaction rules and are therefore susceptible to being discontinued by the following conditions:

- Zone merges (when the merge results in a zoneDB change)
- Zone updates from remote switches
- Forceful transaction aborts by CLI users (that is, `cfgtransabort` *`transaction-token`* operations)

Zoning Checksums
All FOS API REST clients will be expected to store and provide the MD5 (128-bit) ZoneDB checksum in their zoning commit operations. If the provided checksum does not match the checksum stored in zoning, the commit operation will not be allowed. The intent of this requirement is to avoid management interfaces operating off of a zoneDB cache to push in stale data and thus potentially overwrite zone updates that were made after the zoneDB copy was synced.

Diagnostic Port Requests

You can use REST requests to perform Fabric OS ClearLink™ Diagnostics (D_Port) requests:

- D_Port test start
- D_Port test restart
- D_Port test stop

For more information about D_Port configuration and results, see brocade-fibrechannel-diagnostics.

Protocol Support

The Fabric OS REST API supports both the HTTP and HTTPS protocols. The default HTTP port number is 80, and the default HTTPS port number is 443.

NOTE
Although the HTTP protocol is supported, Broadcom strongly recommends using the HTTPS protocol for greater communication security. To use HTTPS, a valid security certificate must be installed on the switch and the HTTPS protocol must be enabled on the switch before beginning REST operations. Refer to the *Brocade Fabric OS Administration Guide* for instructions on both these actions.

NOTE
Broadcom also recommends enabling KeepAlive when using the HTTPS protocol. See Fabric OS REST Session Configuration to enable KeepAlive.

Scalability Recommendations for FOS REST API Clients

In a heavily managed fabric environment, multiple applications may be used to manage a switch at the same time. In such environments, Broadcom recommends not to exceed the total number of application instances with the following guidelines for Gen5 and Gen6 platforms.

These recommendations are based on the default settings described in Fabric OS REST Session Configuration.

Table 1: Total Number of Concurrent Instances of Management Applications

Management Application	Number of Instances	Notes
Brocade SANnav Management Portal	2	The total number of concurrent SANnav and Network Advisor sessions should be no greater than 2.
Brocade Network Advisor	2	
SNMP client	1	—
Brocade Web Tools	1	—
FOS REST API clients	3	—

Best Practices for FOS REST API Clients

The normal response time from a switch is less than 30 seconds for an individual request. If a response to a request is not received in 30 seconds, the FOS REST API client should retry the request up to 3 times. If the switch responds with a 503 Service Unavailable, the FOS REST API client should use the same retry logic described above.

For FOS REST API clients to have predictable and stable performance on Brocade Gen5 platforms, it is recommended to add a one second delay between each request. Utilizing this recommendation for Brocade Gen5 platforms in a large scale fabric minimizes the occurrence of FOS REST API requests that may require a retry. This one second delay is not necessary for Brocade Gen5 platforms in limited scale fabrics.

FOS REST API and Brocade Extension

The FOS REST API supports the Brocade Extension product as implemented in Fabric OS 8.2.0.

Brocade Extension product support includes the following:

- Port configuration and statistics monitoring for Gigabit Ethernet ports
- IP interface configuration
- Route configuration
- IPsec configuration
- Tunnel configuration
- Tunnel statistics monitoring
- Circuit configuration
- Circuit statistics monitoring

Extension-Specific Limitations

- FOS REST API support for Fabric OS Extension products is limited to Brocade 7840 and 7810 Extension switches and Brocade SX6 Extension blades.
- The FOS REST API is not supported on Brocade 7800 switches or Brocade FX8-24 blades.

Restrictions on FOS REST API Configuration Attributes and Statistics

Not all configuration attributes and statistics that are part of the Fabric OS CLI may be exposed through the FOS REST API interface. This is either because of limitations in the FOS REST API or because their values may be obvious due to the way data modeling is done in YANG files.

Examples of this include:

- The ability to set values for attributes (such as setting a Gigabit Ethernet port as a LAN or WAN port) is not exposed through the FOS REST API interface.
- In the CLI version of the `portshow fciptunnel` command output, the “Flags” section identifies the features that are enabled on the tunnel. In the FOS REST API output, this is not displayed, becaues these attributes are modeled individually in each YANG file.

Using the Brocade FOS REST API

The following items should be kept in mind when using the Brocade FOS REST API.

Before You Begin

Before you can use the FOS REST API, you must obtain a user name and password authorized to access Fabric OS through the FOS REST API. The FOS REST API allows you to log in using basic authorization (user name and password) or an encrypted authorization token (base64 encoded user-name:auth-token). For more information about encrypted authorization, see Generating an Encrypted Authorization Token.

> **NOTE**
> If your user password is expired, FOS REST API allows you to change your password before you log in to the REST API (see *Configuring a User Password*).

To use the recommended HTTPS protocol, a valid security certificate must be installed on the switch and the HTTPS protocol must be enabled on the switch before beginning REST operations. Refer to the *Brocade Fabric OS Administration Guide* for instructions on both these actions.

Logging In

To log in to a device, you must provide a valid Fabric OS user name and password or encrypted authorization token through an authorization header in a `POST https://<device_ID>/rest/login` request. The device_ID can be in the form of either an IPv4 or IPv6 address or a host:port ID. If the authentication is successful, an authorization key is returned to the client in the response authorization header. Subsequent FOS REST API operations must include this authorization key in the request authorization header. The client applications use this token to obtain further access to the server using the persistent connection. For examples of logging in and out, see *Logging On, Retrieving Switch Information, and Logging Out* and *Logging On, Retrieving Chassis Information, and Logging Out Using Session-Less REST API Operation*.

Examples

Here are two forms of login statements.

```
POST https://10.10.10.10/rest/login
POST https://[10:10:10:eb:1a:b7:77:bc]:443/rest/login
```

URI Headers

For JSON, in addition to the Authorization key, you must include the following keys and values in the headers:

- Accept = application/yang-data+json
- Content-Type = application/yang-data+json

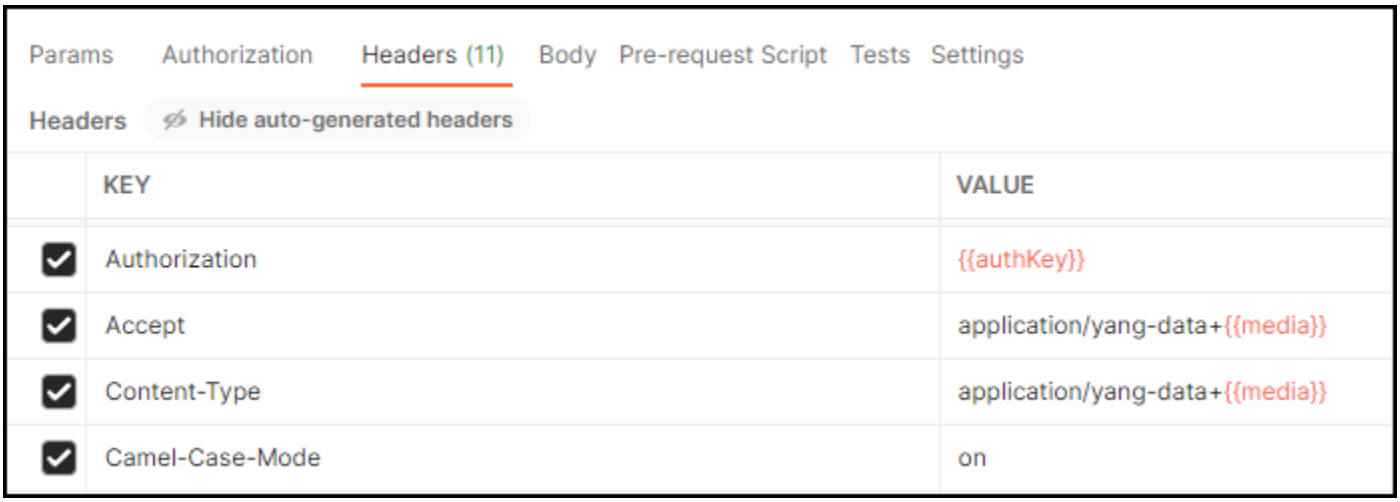
For camelCase, you must include the following key and value in the headers:

- Key – Camel-Case-Mode
- Value – on

For kebab-case, make sure that Camel-Case-Mode is not included in the headers or include the following key and value in the headers:

- Key – Camel-Case-Mode
- Value – off

Figure 2: URI Headers



URI Request

```
POST https://10.10.10.10/rest/login
```

Request Body

There is no request body; however, you must provide a valid Fabric OS user name and password (such as, administrator / password) through an authorization header.

Request Response

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the header.

```
<?xml version="1.0"?>
<Response>
    <switch-parameters>
        <user-name>admin</user-name>
        <chassis-access-role>admin</chassis-access-role>
        <home-virtual-fabric>128</home-virtual-fabric>
        <virtual-fabric-role-ids>
            <role-id>admin=1-128</role-id>
        </virtual-fabric-role-ids>
        <virtual-fabric-ids>
            <fabric-id>77</fabric-id>
            <fabric-id>128</fabric-id>
        </virtual-fabric-ids>
        <virtual-fabric-supported>true</virtual-fabric-supported>
        <virtual-fabric-enabled>true</virtual-fabric-enabled>
        <firmware-version>v9.1.0</firmware-version>
        <model>173.1</model>
    </switch-parameters>
</Response>
```

If authentication is successful, a session authorization key (for example, Authorization →Custom_Basic Tk0ZmY2Zjg3NTY2ZDYwYjhmNj5NGQ0NTkzZjM0M2ZlMWM=) is returned to the client in the response headers.

Subsequent FOS REST API operations must include this authorization key in the request authorization header. The client applications use this token to obtain further access to the server using the persistent connection.

Login Response

The REST login response provides user and switch configuration details to allow you to easily discover the switch with all of the logical switches. The REST login response includes the following user and switch configuration details:

- User name – The name of the account used at login.
- User role – The user role for the switch in non-VF mode.
- Chassis access role – The account's access permissions regarding chassis-level commands.
- Home Virtual Fabric – The account's home Virtual Fabric. If not configured in the ldap-role-map, it returns a '0' value.
- Virtual Fabric-enabled – Whether Virtual Fabrics is enabled on the chassis.
- Virtual Fabric-supported – Whether Virtual Fabrics is supported on the chassis.
- Days to user password expiration – The number of day until the user password expires. If the password for the logged in user does not expire, this value is not included in the login response.
- Firmware version – The active firmware version running on the switch.
- Model – The model name of the switch.
- Virtual Fabric role ID – The Virtual Fabrics roles and IDs assigned to the user account.
- Virtual Fabric ID – The virtual fabric ID (VFID) of the logical switch for the logged in user.

Logging Out

To log out from a device, close the session using a `POST https://<device_ID>/rest/logout` request. You must include the session authorization key in the request authorization header. The device_ID can be in the form of either an IPv4 or IPv6 address or a host:port ID.

Examples

Here are two forms of logout statements.

```
POST https://10.10.10.10/rest/logout
POST https://[10:10:10:eb:1a:b7:77:bc]:443/rest/logout
```

URI Headers

For JSON, you must include the following keys and values in the header.

- Accept = application/yang-data+json
- Content-Type = application/yang-data+json

URI Request

```
POST https://10.10.10.10/rest/logout
```

Request Body

There is no request body; however, you must include the session authorization key in the request authorization header.

Request Response

There is no request response. When the operation is successful, the response contains an empty message body and a “204 No Content” status in the header.

Password Change Before REST API Login

If your user password is expired, FOS REST API allows you to change your password before you log in to the REST API (see Configuring a User Password in the brocade-security Examples section).

REST Session Count Limit

The default limit for FOS REST API sessions is three sessions, but the maximum number of sessions is configurable from 1 to 10. Be aware that increasing the number of FOS REST API sessions or the number of requests within a session may negatively affect latency in a fabric. Refer to REST session configuration for details on setting the session count limit.

Character Restrictions

The following character restrictions apply when making FOS REST API requests:

- Key values containing the “forward slash” reserved character (/) in the URI must be percent-encoded per RFC 8040 and RFC 3986 as “%2f”.
- Key values containing a comma (,), which is used when there is more than one key, must be encoded as “%2c”.
- The following lists common characters that are used in XML (per RFC 3470) and must be properly encoded to be used in strings:
  - The less than character (<) must be encoded as "<".
  - The ampersand character (&) must be encoded as "&".
  - The greater than character (>) must be encoded as ">".
  - The double quote character (") must be encoded as """.
  - The apostrophe character (') must be encoded as "'".
- The following lists common characters that are used in JSON (per RFC 7159) and must be properly encoded to be used in strings:
  - The double quote character (") must be encoded as \".
  - The single quote character (') must be encoded as \'.
  - The backslash character (\) must be encoded as \\.

Keyword Support

The FOS REST API supports standard YANG keywords such as `if` , `when` , and `mandatory` . Refer to the individual module .yang files to see where these keywords are supported; they are not identified in this document.

Configuration Upload and Download

Fabric OS REST configurations can be saved and restored using the standard **configUpload** and **configDownload** commands as well as during a high availability (HA) failover operation. If you downgrade a device to a version of Fabric OS earlier than 8.2.0, FOS REST API is not supported and does not function, and any `mgmtapp` configuration commands are discarded.

Using Brocade FOS REST API Session-Less Operation

The following items should be kept in mind when using FOS REST API session-less operation.

Before You Begin

Before you can use FOS REST API session-less operation, you must obtain a user name and password authorized to access Fabric OS through FOS REST API session-less operation. To use the recommended HTTPS protocol, a valid HTTPS security certificate must be installed on the switch and the HTTPS protocol must be enabled on the switch before beginning REST HTTPS operations. Refer to the *Brocade Fabric OS Administration Guide* for instructions on both these actions. FOS REST API session-less operation follows the same role-based access control restrictions and constraints as FOS REST API (see Security and the FOS REST API).

NOTE
Fabric OS 8.2.2 or later supports FOS REST API session-less operation for GET requests only.

NOTE
You must have at least one REST API session available to execute session-less requests.

Logging In and Out Using FOS REST API Session-Less Operation

A FOS REST API session-less operation request completes the login, GET operation, and logout as one complete request. You can only use HTTP Basic Authentication formats for FOS REST API session-less operation, which includes plain text or Base64. The following shows example GET request structures using either plain text or Base64-encoded text:

- **Plain Text Authentication Structure**

```
curl –v –X GET -u <user_name>:<password> -H "<media_type>" "<base_URI>/<FOS_REST_API_request>"
```

- **Base64 Authentication Structure**

```
curl –v –X GET -H "Authorization:Basic <Base64_authentication>" -H "<media_type>"
 "<base_URI>/<FOS_REST_API_request>"
```

FOS REST API session-less operation logout occurs automatically when the request is complete.

Plain Text Authentication Example

This example uses a GET request with plain text authentication to retrieve information about the chassis in JSON format.

Structure

```
curl –v –X GET -u <user_name>:<password> -H "<media_type>" "<base_URI>/rest/running/brocade-chassis/chassis"
```

Request

```
curl -v -X GET -u admin:password1 -H "Accept: application/yang-data+json" "https://10.11.12.13/rest/running/
brocade-chassis/chassis"
```

Response Body


```
{
 "Response": {
  "chassis": {
   "chassis-user-friendly-name": "sanchassis1",
   "license-id": "10:00:c1:f2:3c:00:af:40",
   "chassis-wwn": "10:00:c1:f2:3c:00:af:4f",
   "serial-number": "FER0304N00Z",
   "manufacturer": "Brocade Communications Systems LLC",
   "part-number": "40-1001199-03",
   "vf-enabled": true,
   "vf-supported": true,
   "fcr-enabled": true,
   "fcr-supported": true,
   "max-blades-supported": 1,
   "vendor-revision-number": "",
   "vendor-part-number": "",
   "vendor-serial-number": "",
   "product-name": "g630",
   "date": "12/24/2020-10:34:37"
  }
 }
```

Base64 Authentication Example

This example uses a GET request with Base64 authentication to retrieve information about the chassis in JSON format.

Structure

```
curl –v –X GET -H "Authorization:Basic <Base64_authentication>" -H "<media_type>" "<base_URI>/rest/running/
brocade-chassis/chassis"
```

Request

```
curl -v -X GET -H "Authorization:Basic YWRtaW46cGFzc3dvcmQx" -H "Accept: application/yang-data+json"
 "https://10.11.12.13/rest/running/brocade-chassis/chassis"
```

Response

```
{
 "Response": {
  "chassis": {
   "chassis-user-friendly-name": "sanchassis1",
   "license-id": "10:00:c1:f2:3c:00:af:40",
   "chassis-wwn": "10:00:c1:f2:3c:00:af:4f",
   "serial-number": "FER0304N00Z",
   "manufacturer": "Brocade Communications Systems LLC",
   "part-number": "40-1001199-03",
   "vf-enabled": true,
   "vf-supported": true,
   "fcr-enabled": true,
   "fcr-supported": true,
   "max-blades-supported": 1,
   "vendor-revision-number": "",
   "vendor-part-number": "",
   "vendor-serial-number": "",
   "product-name": "g630",
   "date": "12/24/2020-10:44:34"
  }
 }
```

FOS REST API Role-Based Access Control

The FOS REST API supports user-created roles and permissions. For instructions on managing roles, refer to the *Brocade Fabric OS Administration Guide*.

To view a list of FOS REST API roles, enter `mgmtapp --show rbacinfo`.

```
sw0:admin> mgmtapp --show rbacinfo

-------------------------------------------------------------------------------------------------------------------
URL                                                                RBAC                          CONTEXT_TYPE
-------------------------------------------------------------------------------------------------------------------
/rest/auth-token                                                   None                             both
/rest/brocade-module-version                                       None                             both
/rest/login                                                        None                             both
/rest/logout                                                       None                             both
/rest/modules-state                                                None                             both
```

```
/rest/operations/device-management                    NxPortManagement          both
/rest/operations/extension                            SwitchPortManagement      chassis
/rest/operations/fibrechannel-fabric                  Fabric                    vf
/rest/operations/fibrechannel-zone                    Zoning                    vf
/rest/operations/firmwaredownload                     FirmwareManagement        both
/rest/operations/license                              License                   both
/rest/operations/lldp                                 DCE                       both
/rest/operations/pcie-health-test                     Blade                     chassis
/rest/operations/show-status                          None                      both
/rest/operations/supportlink                          SessionManagement         both
/rest/operations/supportsave                          Restricted                both
/rest/running/brocade-access-gateway/device-list      AG                        vf
/rest/running/brocade-access-gateway/f-port-list      AG                        vf
(output truncated)
```

Table 2: RBAC Output

Output	Description
URL	The request URI, which is the URI used to perform a GET, POST, PATCH, DELETE, HEAD, or OPTIONS operation.
RBAC	The meta-object format (MOF) class. The REST requests are grouped into feature sets called MOF classes. For example, /rest/operations/fibrechannel-zone and and /rest/running/brocade-zone/alias are all related under zoning and are therefore grouped together under a MOF class called **Zoning**.
CONTEXT_TYPE	The Virtual Fabric or chassis context type or both. REST requests can be issued in one or more of the contexts. Note that issuing of a chassis REST request requires chassis permissions.

HTTP Header

HTTP header fields are components of the message header of a request and response in HTTP. They define the operating parameters and are name-value pairs that appear in both request and response messages.

The following table contains the supported HTTP methods for the media types.

Table 3: Supported HTTP Methods and Media Types

Method	Supported Media Types
DELETE	application/yang-data+xml (default) and application/yang-data+json
GET	application/yang-data+xml (default) and application/yang-data+json
HEAD	application/yang-data+xml (default) and application/yang-data+json
OPTIONS	All media types for this method.
POST	application/yang-data+xml (default) and application/yang-data+json
PATCH	application/yang-data+xml (default) and application/yang-data+json

For more information about supported media types, refer to media types.

Request Headers

The FOS REST API supports only the following request headers. All requests that use a body must have the body in XML format.

- Accept
- Authorization
- Cache-Control
- Content-Length
- Date
- Host
- User-Agent

NOTE
All FOS REST API requests that return data return that data in XML or JSON format.

Media Types

In REST, “Media type” is an application-specific format with a well-defined name represented in the form of an identifier used to identify the form of the data contained within a resource representation.

Media types are specified in the Accept header for requests and in the Content-Type header for responses. Media types are resource-specific, this allows them to change independently and support formats that other resources do not. The FOS REST API supports the XML (default) and JSON media types. If you do not specify Accept or Content-Type headers, the media type defaults to XML.

For the JSON media type, you must include the following keys and values in the headers:

- Accept = application/yang-data+json
- Content-Type = application/yang-data+json

HTTP Status Codes and Messages

Both success and error status conditions are reported to the client by way of the HTTP status line, which contains the HTTP status code. These application-specific messages are similar to CLI responses.

The following table lists and describes the HTTP status codes supported by the Fabric OS REST API.

Table 4: Supported HTTP Status Codes

Status Line Value	Description
100 Continue	POST is accepted, 201 should follow
200 OK	Success with response body
201 Created	Request successfully created a resource
202 Accepted	Request to create a resource accepted
204 No Content	Success without a response body
400 Bad Request	Invalid request message
403 Forbidden	Access to resource denied
404 Not Found	Resource target or resource node not found
405 Method Not Allowed	Method not allowed for target resource

Status Line Value	Description
413 Request Entity Too Large	The request entity is too large
414 Request URI Too Large	The request URI is too large
415 Unsupported Media	Not a supported media type
500 Internal Server Error	Operation failed. In this case, the response body will contain the application's specific error message
501 Not Implemented	Unknown operation
502 Server Busy	Server busy error
503 Service Unavailable	Recoverable server error

Supported Methods

The FOS REST API supports the DELETE, GET, HEAD, OPTIONS, PATCH, and POST methods.

DELETE Method

The DELETE method is used to delete a specified resource.

The latest update to the HTTP 1.1 specification (RFC 7231) explicitly permits an entity body in a DELETE request: A payload within a DELETE request message has no defined semantics; sending a payload body on a DELETE request might cause some existing implementations to reject the request.

The following example shows a DELETE operation to remove an existing IP interface.

Request URI

```
DELETE https://10.10.10.10/rest/running/brocade-interface/extension-ip-interface
```

Request Body

```
<extension-ip-interface>
   <name>4/5</name>
   <dp-id>0</dp-id>
   <ip-address>10.10.10.05</ip-address>
   <ip-prefix-length>24</ip-prefix-length>
</extension-ip-interface>
```

Request Response

When the operation is successful, the response contains an empty message body and a "204 No Content" status in the header.

Notes

- An authorization header is required to perform a DELETE operation.
- A request payload is required when any zone members are deleted.

GET Method

The GET method is used to retrieve the representation of the resource (for example, base configuration) including the metadata information.

NOTE
The GET operation only depends on key leafs in the request body and the key leaf values as per the yang. If any non-key leafs are included in the request body along with key leafs, the non-key leafs are ignored and no

validation is performed for the non-key leaf values. The GET response body always contains the actual values from the switch at any time irrespective of what is sent in the request body for non-key leafs.

The following example shows a GET operation to display information about the switch you are logged in to.

Request URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-switch/fibrechannel-switch/
```

Request Body

No request body is required.

NOTE
A request payload is not required for a GET operation.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-switch>
        <name>10:00:a1:b1:1c:1d:ac:1c</name>
        <domain-id>43</domain-id>
        <fcid>16776235</fcid>
        <fcid-hex>0xfffc2b</fcid-hex>
        <user-friendly-name>66-243_X6-4_Venator</user-friendly-name>
        <enabled-state>2</enabled-state>
        <is-enabled-state>true</is-enabled-state>
        <operational-status>2</operational-status>
        <banner/>
        <up-time>73389</up-time>
        <domain-name>brm.bsnlab.broadcom.net</domain-name>
        <dns-servers>
            <dns-server>190.19.190.10</dns-server>
            <dns-server>190.19.190.20</dns-server>
        </dns-servers>
        <principal>0</principal>
        <ip-address>
            <ip-address>10.30.40.50</ip-address>
        </ip-address>
        <subnet-mask>255.255.10.20</subnet-mask>
        <model>165.0</model>
        <firmware-version>v9.0.0</firmware-version>
        <ip-static-gateway-list>
            <ip-static-gateway>10.20.30.1</ip-static-gateway>
        </ip-static-gateway-list>
        <vf-id>-1</vf-id>
        <fabric-user-friendly-name>East</fabric-user-friendly-name>
        <ag-mode>0</ag-mode>
        <dynamic-load-sharing>lossless-dls</dynamic-load-sharing>
        <in-order-delivery-enabled>false</in-order-delivery-enabled>
        <advanced-performance-tuning-policy>exchange-based</advanced-performance-tuning-policy>
        <lacp-system-mac-address>c4:f5:7c:59:ac:6e</lacp-system-mac-address>
```

```
        <lacp-system-priority>32768</lacp-system-priority>
    </fibrechannel-switch>
</Response>
```

HEAD Method

The HEAD method retrieves the specified resource's metadata.

Request URI

```
HEAD https://10.10.10.10/rest/running/brocade-fibrechannel-switch/fibrechannel-switch/
```

Request Body

No request body is required.

Request Response

On successful retrieval of the metadata, the response contains an empty message body and a "200 OK" status in the headers.

> **NOTE**
> A request payload is not required for a HEAD operation.

OPTIONS Method

The OPTIONS method is used to retrieve the allowed methods on the resource identified by the given request. The response to this operation contains the headers and an empty response body. The "Allow" header in the response contains the allowed operations on the resource.

For an OPTIONS request, the leaf elements within a second-level container return all the supported methods for the whole container. For example, in the brocade-fibrechannel-diagnostics module, "fec" is a container that has three leafs: "enable", "options" and "active". Only the "enable" leaf is read-write; the other two leafs are read-only. Nonetheless, when an OPTIONS request is made that includes the "active" or "options" leafs, the request response body will list read-write methods (such as PATCH), even though read-write methods are not valid for those leafs.

Request URI

```
OPTIONS https://10.10.10.10/rest/running/brocade-zone/defined-configuration
```

Request Body

No request body is required.

Request Response

The response contains an empty message body and a "Allow:" line in the header. This lists the supported operations for the specified module.

```
HTTP/1.1 200 OK
Allow: DELETE, GET, HEAD, POST, PATCH     <= Allowed operations
Cache-Control: no-cache
Connection: close
Content-Secure-Policy: default-src 'self'
Content-Type: application/yang-data+xml
Date: Wed, 18 Oct 2017 20:53:32 GMT
Server: Apache
Transfer-Encoding: chunked
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
```

```
X-XSS-Protection: 1; mode=block
```

Notes

- A request payload is not required for an OPTIONS operation.
- The OPTIONS method is supported at the container and list levels but is not supported at the leaf level.

PATCH Method

The PATCH method is used to edit or update the leaf attributes of the resource (List or Container), if the system supports the modification. For example, modifying the leaf or list child resource of an access control list (ACL) “sequence” command is not possible, as it is not allowed by the system.

> **NOTE**
> Not all Fabric OS REST API modules guarantee atomicity of PATCH operations. If a PATCH operation returns an error status code, it is recommended that the application issue a GET request for the same node(s) in order to retrieve and compare the state of the node(s) after the failed PATCH operation.

The FOS REST API uses the PATCH method (RFC 8072) which has features not possible with the “plain-patch” mechanism defined in RESTCONF (RFC 8040).

- It allows multiple sub-resources to be edited within the same PATCH method.
- It allows a more precise editing operation than the “plain-patch” mechanism.
- It uses an “edit” list with an explicit processing order. These edits are processed in the client-specified order, and error processing can be precise even when multiple errors occur in the same PATCH request.
- A request payload is required for a PATCH method request, except when the configuration value can be supplied in the URI, as shown in the “request in body” example below.
- Because a PATCH operation overwrites the entire object, all desired content must be included in the request, including any content that was already in the leaf, list, or container. If you do not include all content, the operation fails with an "400 Bad Request" response.

The following example shows two methods to make a PATCH request to update the switch user-friendly-name to “My_switch”.

Request URI (for a request in the URI)

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-switch/fibrechannel-switch/
name/10:10:14:f5:7c:46:3d:40/user-friendly-name/My_switch
```

Request URI (for a request in the body)

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-switch/fibrechannel-switch/
name/10:10:14:f5:7c:46:3d:40/
```

Request Body (for a request in the body)

```
<fibrechannel-switch>
   <user-friendly-name>My_switch</user-friendly-name>
</fibrechannel-switch>
```

Request Response

When the operation is successful, the response contains an empty message body and a “204 No Content” status in the header.

POST Method

The POST method allows you to create a new resource in the resource location identified by the URI specified in the request.

> **NOTE**
> Not all Fabric OS REST API modules guarantee atomicity of POST operations. If a POST operation returns an error status code, it is recommended that the application issue a GET request for the same node(s) in order to retrieve and compare the state of the node(s) after the failed POST operation.

The following example shows a POST method request to create a new zone.

Request Header

```
POST https://10.10.10.10/rest/running/brocade-zone/defined-configuration
```

Request Body

```
<defined-configuration>
   <zone>
      <zone-name>newZone1</zone-name>
      <zone-type>0</zone-type>
      <member-entry>
         <entry-name>10:10:20:21:f8:f0:3a:10</entry-name>
         <entry-name>10:10:20:21:f8:f0:38:01</entry-name>
      </member-entry>
   </zone>
</defined-configuration>
```

Request Response

When the operation is successful, the response contains an empty message body and a “201 Created” status in the header.

> **NOTE**
> A request payload is required for a POST request, except in those cases where the configuration value is supplied in the URI.

Query Parameter Support

You can use a query parameter to filter or search objects returned via HTTP REST responses. The query string is sent in the URI after the question mark (?) delimiter and is usually in name=value pair formats.

> **NOTE**
> The query parameter is not supported by all modules. Query parameter support is indicated in the individual modules.

The query parameter is supported only on second-level container URI requests. If there is no entry present in the database, it returns an empty list.

Note that the reserved characters ($ & + , / : ; = ? @) are not allowed in a query. You must be percent-encode the reserved characters as per the standards (RFC 8040 and RFC 3986). For additional information on HTML URI encoding values, refer to https://en.wikipedia.org/wiki/Percent-encoding.

Examples

This example uses a GET request to view a nameserver with a valid node name.

```
GET <base_URI>/
running/brocade-name-server/fibrechannel-name-server?node-name=20%3A04%3A00%3A11%3A0d%3A75%3A00%3A09
```

This example uses a GET request to view a nameserver with a valid node name and VF ID.

```
GET <base_URI>/
running/brocade-name-server/fibrechannel-name-server?vf-id=10&node-name=20%3A04%3A00%3A11%3A0d%3A75%3A00%3A09
```

FOS REST API Version History

The FOS REST API supports resource-API and module-level versioning using the brocade-module-version module.

You can use the resource API version history to determine backward compatibility.

- Major number – A backward incompatible API change is made (such as, removing a deprecated top-level container).
- Minor number – A backward compatible API change is made (such as, adding a top-level container name or leaf).
- Patch – A backward compatible API change is made (such as, new pattern inclusion for a leaf).

Resource API Version

The resource API version displays top-level URI changes for major, minor, and patch releases. The resource API version displays the current API release in the 'Content-Type' response header. For example, Content-Type →application/yang-data+xml;version=1.70.0.

For example, the resource API version for Fabric OS 9.2.0 is 1.70.0 (major.minor.patch).

Module Level Version

The module level version displays the revision numbering for the individual modules. Module level versioning is updated for changes to leafs or objects within an individual module.

The following table details the module level version for each REST API module in Fabric OS 9.2.x.

Table 5: Module Level Version

Module Name	Module Version
Auth-token	1.0.0
brocade-access-gateway	1.40.0
brocade-chassis	1.40.0
brocade-extension	3.10.0
brocade-extension-ip-route	1.20.0
brocade-extension-ipsec-policy	1.10.0
brocade-extension-tunnel	3.0.0
brocade-fabric	2.20.0
brocade-fabric-traffic-controller	1.20.0
brocade-fdmi	1.30.0
brocade-fibrechannel-configuration	2.10.0
brocade-fibrechannel-diagnostics	3.0.0
brocade-fibrechannel-logical-switch	2.30.0
brocade-fibrechannel-routing	2.0.0

brocade-fibrechannel-switch	3.40.0
brocade-fibrechannel-trunk	1.20.0
brocade-ficon	1.0.0
brocade-firmware	2.0.0
brocade-fru	2.10.0
brocade-interface	3.40.0
brocade-license	1.30.0
brocade-lldp	1.0.10
brocade-logging	2.0.0
brocade-management-ip-interface	1.10.0
brocade-maps	3.0.0
brocade-media	2.20.0
brocade-module-version	1.0.0
brocade-name-server	2.10.0
brocade-operation-configdownload	1.0.0
brocade-operation-configupload	1.0.0
brocade-operation-date	1.0.0
brocade-operation-device-login-rebalance	1.0.0
brocade-operation-device-management	1.0.0
brocade-operation-diagnostics	1.0.0
brocade-operation-extension	1.0.10
brocade-operation-fabric	1.0.0
brocade-operation-factory-reset	1.10.0
brocade-operation-firmwaredownload	1.10.0
brocade-operation-license	1.20.0
brocade-operation-lldp	1.10.0
brocade-operation-management-ethernet-interface	1.0.0
brocade-operation-pcie-health	1.0.10
brocade-operation-port	1.10.0
brocade-operation-portchannel	1.0.0
brocade-operation-port-decommision	1.0.0
brocade-operation-reboot	1.0.0
brocade-operation-restart	1.0.0
brocade-operation-sdd-quarantine	1.0.0
brocade-operation-security-acl-policy	1.0.0
brocade-operation-security-authentication-configuration	1.0.0
brocade-operation-security-authentication-secret	1.0.0
brocade-operation-security-certificate	1.0.10

brocade-operation-security-fabric-wide-policy-distribute	1.0.0
brocade-operation-security-ipfilter	1.0.0
brocade-operation-security-policy-chassis-distribute	1.0.0
brocade-operation-security-policy-distribute	1.0.0
brocade-operation-security-reset-violation-statistics	1.0.0
brocade-operation-security-role-config	1.0.0
brocade-operation-show-status	1.10.0
brocade-operation-supportlink	1.0.0
brocade-operation-supportsave	2.0.0
brocade-operation-time	1.0.0
brocade-operation-traffic-optimizer	1.10.0
brocade-operation-usb-delete-file	1.0.10
brocade-operation-zone	1.20.0
brocade-security	4.0.0
brocade-snmp	2.0.0
brocade-supportlink	1.10.10
brocade-time	1.20.10
brocade-traffic-optimizer	1.10.0
brocade-usb	1.10.0
brocade-zone	2.10.30
login	1.20.0
logout	1.0.0

Requesting the Module Level Versions

The following example uses a GET request to retrieve the module level version information.

Structure

`https://<device_ID>/rest/brocade-module-version`

Request URI

```
GET https://10.10.10.10/rest/brocade-module-version
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <brocade-module-version>
        <module>
```

```
            <name>login</name>
            <version>1.20.0</version>
            <uri>/rest/login</uri>
            <objects/>
        </module>
        <module>
            <name>logout</name>
            <version>1.0.0</version>
            <uri>/rest/logout</uri>
            <objects/>
        </module>
...
        <module>
            <name>brocade-snmp</name>
            <version>1.20.0</version>
            <uri>/rest/running/brocade-snmp</uri>
            <objects>
                <object>system</object>
                <object>mib-capability</object>
                <object>trap-capability</object>
                <object>v1-account</object>
                <object>v1-trap</object>
                <object>v3-account</object>
                <object>v3-trap</object>
                <object>access-control</object>
            </objects>
        </module>
        <module>
            <name>brocade-operation-supportsave</name>
            <version>1.10.10</version>
            <uri>/rest/operations/supportsave</uri>
            <objects>
                <object>connection</object>
            </objects>
        </module>
    </brocade-module-version>
</Response>
```

FOS REST API and Brocade Virtual Fabrics

FOS REST API operations can occur across virtual fabrics.

When you log on to a switch as described in Logging In, by default the session is logged on to the logical switch that belongs to the “Home” virtual fabric of the REST user. If you want to execute a FOS REST API request on other logical switches, you must include the virtual fabric ID (vf_id) as part of the request query parameter.

This example uses a GET request to retrieve the switch information for a switch in the virtual fabric with an VFID of 10.

Structure

```
GET <base_URI>/running/brocade-fabric/fabric-switch?vf-id=<vf-id#>
```

Request URI

```
GET https://10.10.10.10/rest/running/brocade-fabric/fabric-switch?vf-id=10
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-switch>
        <name>10:10:10:40:1a:b7:47:bc</name>
        <chassis-wwn>10:10:10:40:7c:4a:ac:eb</chassis-wwn>
        <domain-id>91</domain-id>
        <fcid>16776283</fcid>
        <fcid-hex>0xfffc5b</fcid-hex>
        <switch-user-friendly-name>DCX_066_240</switch-user-friendly-name>
        <chassis-user-friendly-name>BrocadeDCX</chassis-user-friendly-name>
        <firmware-version>v9.0.1</firmware-version>
        <vf-id>10</vf-id>   <== Virtual fabric ID
        <ip-address>10.10.10.91</ip-address>
        <fcip-address>0.0.0.0</fcip-address>
        <ipv6-address>2620:100:4:fa01:207:f8ff:fe0f:b9f0</ipv6-address>
        <principal>0</principal>
    </fibrechannel-switch>
</Response>
```

Fabric OS REST Session Configuration

The `apploginhistory` and `mgmtapp` Fabric OS CLI commands allow you to manage REST session-related operations.

The `mgmtapp` command allows you to perform REST session-related operations, including enabling, disabling, and terminating the REST session, as well as configuring the maximum number of REST sessions for an entire switch or chassis, configuring keepalive for the REST session, and setting throttling values. On chassis-based systems, this command is only supported on the active CP.

Entering `mgmtapp --show` displays the number of active REST sessions and the REST throttling configuration values (sample request counts, sample time (in seconds), idle time (in seconds), whether keepalive is enabled, and keepalive timeout (in seconds)). The following example shows the output of the command when all values are at their defaults.

```
switch:admin> mgmtapp --show
REST Interface State: Enabled
REST Session Count: 3
REST Throttling Configurations:
   Sample Requests      : 30
   Sample Time (in sec) : 30
   Idle Time (in sec)   : 3
   KeepAlive : Enabled
   KeepAliveTimout : 15sec
```

NOTE
REST operations are not permitted if the REST interface is disabled in CLI, and are denied after changes to a user account with an existing REST session.

Enabling and Disabling the Fabric OS REST Interface

The Fabric OS REST interface is enabled by default.

To disable the Fabric OS REST interface, enter `mgmtapp --disable REST`.

To re-enable the Fabric OS REST interface, enter `mgmtapp --enable REST`.

Encrypted Authorization Token

Use the Fabric OS CLI to generate an encrypted authorization token that can then be used to log on to the REST API interface. This authorization token is a 256-bit key encrypted with the AES-256 encryption algorithm and is stored persistently in the configuration. Once you base64-encode the authorization token along with your user name (user-name:auth-token), the FOS REST API uses this authorization token for subsequent logons in a base64 format of <user-name:auth-token>.

To log in to the REST API interface using an encrypted authorization token, you must use a custom authorization type (`Custom_Auth`). The HTTP authorization token format is `Custom_Auth <base64-encoded-authorizationtoken>` . The custom authorization type replaces the standard basic authorization used to log in to the REST API interface.

The FOS REST API authorization key is supported for local user accounts only. Note that the following changes to the user account remove the encrypted authorization key from the configuration:

- User account deleted
- User account changed (such as, role)
- User password changed
- User password expired
- User account disabled
- User permission revoked
- Factory reset
- User Virtual Fabric modified

Generating an Encrypted Authorization Token

You must use the Fabric OS CLI to generate an encrypted authorization token.

1. To create an encrypted authorization token, enter `mgmtapp --create authtoken` .
2. To display the encrypted authorization token for the current user, enter `mgmtapp --show authtoken` .

```
SU5lZWRTb21lYm9keShIZWxwISlOb3RKdXN0QW55Ym9keShIZWxwISlZb3VLbm93SU5lZWRTb21lb25laGVscCFJTm
V2ZXJOZWVkZWRBbnlib2R5J3NIZWxwSW5BbnlXYXlidXROb3dUaGVzZURheXNBcmVHb25lLEknbU5vdFNvU2VsZk
Fzc3VyZWRCdXROb3dUaGVzZURheXNBcmVHb25lKUFuZE5vd0lGaW5kKU5vd0lGaW5kSSd2ZUNoYW5nZWRNeU1
pbmRBbmRPcGVuZWRVcFRoZURvb3JzaGVscE1lSWZZb3VDYW4sSSdtRmVlbGluZ0Rvd24oSGVscCEp==
```

3. Base64-encode your user name and the encrypted authorization token using the following format: `<user-name:auth-token>` .
   For example, the following user name and the encrypted authorization token (`<user-name:auth-token>`):

```
gumbeaux504:SU5lZWRTb21lYm9keShIZWxwISlOb3RKdXN0QW55Ym9keShIZWxwISlZb3VLbm93SU5lZWRTb21lb
25laGVscCFJTmV2ZXJOZWVkZWRBbnlib2R5J3NIZWxwSW5BbnlXYXlidXROb3dUaGVzZURheXNBcmVHb25lLEknbU
5vdFNvU2VsZkFzc3VyZWRCdXROb3dUaGVzZURheXNBcmVHb25lKUFuZE5vd0lGaW5kKU5vd0lGaW5kSSd2ZUNoY
W5nZWRNeU1pbmRBbmRPcGVuZWRVcFRoZURvb3JzaGVscE1lSWZZb3VDYW4sSSdtRmVlbGluZ0Rvd24oSGVscCEp==
```

becomes the following base64-encoded authorization token:

```
Z3VtYmVhdXg1MDQ6SU5lZWRTb211lYm9keShIZWxwISlOb3RKdXN0QW55Ym9keShIZWxwISlZb3VLbm93SU5lZWR
Tb21lb25laGVscCFJTmV2ZXJOZWVkZWRBbnlib2R5J3NIZWxwSW5BbnlXYXlidXROb3dUaGVzZURheXNBcmVHb25l
LEknbU5vdFNvU2VsZkFzc3VyZWQoQnV0Tm93VGhlc2VEYXlzQXJlR29uZSlBbmROb3dJRmluZClOb3dJRmluZEkndm
VDaGFuZ2VkTXlNaW5kQW5kT3BlbmVkVXBUaGVEb29yc2hlbHBNZUlmWW91Q2FuLEknbUZlZWxpbmdEb3duKEh
lbHAhKQ==
```

4. Use the base64-encoded authorization token during the REST login process.

   To log on to the REST API interface using an encrypted authorization token, you must use a custom authorization type (`Custom_Auth`). The HTTP authorization token format is `Custom_Auth <base64-encoded-authorization-token>` . The custom authorization type replaces the standard basic authorization used to log in to the REST API interface.

Enabling and Disabling HTTP Protocol Support

NOTE
Upgrading a switch from a pre-Fabric OS 9.0.0 version to Fabric OS 9.0.0 or later enables the HTTP REST interface by default.

The following example shows the output of the command when all values are at their defaults after an upgrade.

```
switch:admin> mgmtapp --show
REST Configuration:
Interface State: Enabled
Effectivie Protocol: HTTP only
HTTP State: Enabled
Session Count: 3
HTTPS Configuration:
   KeepAlive : Enabled
   KeepAliveTimout : 15sec
```

To enable the HTTP Protocol, enter `mgmtapp --enable rest -protocol http.`

To disable the HTTP Protocol, enter `mgmtapp --disable rest -protocol http.`

The following example shows the output of the command when REST is disabled.

```
G610:admin> mgmtapp --show
  REST Configuration:
        Interface State: Disabled
        Effective Protocol: None
        HTTP State: Disabled
        Session Count: 10

  HTTPS Configuration:
        KeepAlive : Enabled
        KeepAliveTimeout : 15sec
```

Enabling and Disabling Keepalive Mode

The keepalive mode is disabled by default. For HTTPS mode, it is recommended that you enable keepalive mode.

NOTE
Keepalive mode is only supported with HTTPS mode.

To disable the keepalive mode, enter `mgmtapp --disable keepalive`.

To reenable the Fabric OS REST interface, enter `mgmtapp --enable keepalive`.

```
HTTP mode will be disabled after enabling KeepAlive. Do you want to continue? (y or n) y
```

Configuration of Fabric OS REST Interface Session Values

When the Fabric OS REST interface is enabled, it uses the `-maxrestsession` count to determine how many concurrent REST sessions are permitted to an entire switch or chassis. There can be from 1 to 10 REST sessions configured; the default is 3. An error message is returned if you attempt to establish more REST sessions than the permitted number.

In addition to the number of concurrent REST sessions permitted, you can configure the sampling time and idle time in seconds and the sample request count. These options are set using the `mgmtapp --config` *`configuration_parameters`* command.

The configuration parameter is: `-maxrestsession` *`rest session count`*.

```
switch:admin> mgmtapp --config -maxrestsession 4
```

Identification and Termination of Fabric OS REST Sessions

To identify the REST sessions on a platform, enter `mgmtapp --showsessions` to display the number of sessions from the external applications that are currently active. This example shows typical output. In this example, the lines have been broken for clarity, and the REST session line has been rendered in bold.

```
switch:admin> mgmtapp --showsessions
The following are the sessions from the external applications that are active currently:
2021/11/22-08:53:42.719945, 10.10.10.165, admin, SANnavMP220,
 6eb2af2be0da0a661fc2b1fe303fd87aab5338b9d149221fba28d265b304c806, 128, No, HTTPS, d6d3b745-c412-3a7c-9718-
b1f7eebeeaf8
2021/11/22-08:53:43.043335, 10.10.10.165, admin, SANnavMP220,
 ce06efc76b4d7e59a0592c6be367f7841c6bb55236685c18f69702a93afebe66, 128, No, HTTPS, d6d3b745-
c412-3a7c-9718-b1f7eebeeaf8
2021/11/23-16:20:51.878653, 10.10.10.188, admin, PostmanRuntime/7.28.4,
 1d8b145852f867e3584cd4cf8b06e1322c9e503b2202d8e9598238f6fbe3f9d4, 128, No, HTTP, None
2021/11/23-16:22:12.145862, 10.10.10.110, admin, Apache-HttpClient/4.5.12,
 8b7c0d4ce509ef1eb566593ecc75e3b8c68eabfe8c263e145272b9050e582147, 128, No, HTTP, None
```

To terminate a REST session, enter `mgmtapp --terminate` *`session_id`*. You must provide the session ID. This example terminates the REST session identified above.

```
switch:admin> mgmtapp --terminate 1d8b145852f867e3584cd4cf8b06e1322c9e503b2202d8e9598238f6fbe3f9d4
Rest session terminated successfully.
```

Refer to the entry for these commands in the *Brocade Fabric OS Command Reference Manual* for more information and examples.

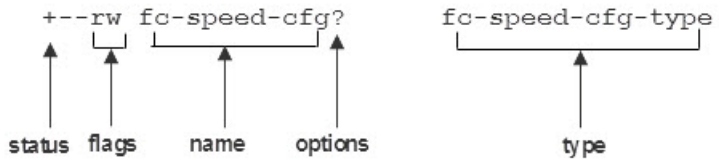
YANG Module Overview

The Fabric OS REST API uses YANG 1.1 (Yet Another Next Generation) as the basis for its data structure.

YANG 1.1 is defined in IETF RFC 7950. It is a data modeling language used to model configuration data, state data, remote procedure calls, and notifications for network management protocols. The Fabric OS REST API YANG files are part of the Fabric OS distribution (/dist/yang.tar.gz).

The following figure shows how individual node fields are typically displayed in a `.yang` tree file.

Figure 3: YANG Module Node Fields



In the tree view of a YANG module, the indents indicate leafs. An asterisk `*` indicates a container for a leaf list, and a question mark `?` indicates an optional item. `ro` indicates a read-only item (that is, the node contains operational data), and `rw` indicates read-write. The `+--` element provides a visual marker for following the tree structure. Refer to the following table for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

YANG node field definitions and possible values are listed in the following table.

Table 6: YANG Node Field Definitions and Values

Field	Definition	Values
status	Status of the node.	+ Current x Deprecated o Obsolete
flags	Type of information.	rw Read-write—The node contains configuration data ro Read-only—The node contains operational data -x RPC statement -n Notification (not supported)
name	The name of the node.	(name)—Choice node *(name)—Case node prefix:name—Augmented from the specified external module
options	The node name can be modified by one of these optional values.	The following optional values can be used: • ?—Optional leaf or presence container • *—Leaf-list • [keys]—Keys for a list
type	The leaf or leaf-list type. If the specified type is other than one of the basic types, then the type is defined in a typedef statement in the `.yang` file.	N/A

Supported Data Types

The following section describes the YANG built-in data types.

The FOS REST API supports additional data types (type definitions, features, and groupings). Additional details about these data types can be found in a data type file (for example, fibrechannel-yang-types.yang) or the associated .yang file (for example, brocade-zone.yang). These files are located in the same folder as the module .yang files. For more information about the additional data types, see Definitions.

YANG Built-In Data Types

The following data types are inherent to the YANG architecture. Refer to RFC 6991 or RFC 2578 for additional details.

binary	Any binary data.
bits	A set of bits or flags.
boolean	Either the value “true” or the value “false”.
counter32	A non-negative integer that monotonically increases until it reaches a maximum value of 2^32-1 (4294967296 in decimal), at which point it wraps around and starts increasing again from zero. It has no preset initial value.
counter64	A non-negative integer that monotonically increases until it reaches a maximum value of 2^64-1 (18446744073709551615 in decimal), at which point it wraps around and starts increasing again from zero. It has no preset initial value.
empty	A leaf that does not have any value.
enumeration	One of an enumerated (explicitly listed) set of strings.
int8	An 8-bit signed integer.
int16	A 16-bit signed integer.
int32	A 32-bit signed integer.
int64	A 64-bit signed integer.
leafref	A reference to a leaf instance.
mac-address	An IEEE 802.xx media access control address (MAC) address. The canonical representation uses lowercase characters.
string	A series of alphanumeric and punctuation characters including spaces treated as a single unit. A string may also include special characters. For a formal definition, refer to http://www.w3.org/TR/2000/WD-xml-2e-20000814 and section 3.2.1 of http://www.w3.org/TR/xmlschema-2/.
timetick	The time (measured in hundredths of a second) between two epochs. When a schema node is defined that uses this type, the description of the schema node identifies both of the reference epochs.
uint8	An 8-bit signed unsigned integer.
uint16	A 16-bit signed unsigned integer.
uint32	A 32-bit signed unsigned integer.
uint64	A 64-bit signed unsigned integer.
union	A data type that is composed of multiple data types.
zero-based-counter32	A counter32 that has zero as its defined initial value.
zero-based-counter64	A counter64 that has zero as its defined initial value.

FOS REST API YANG Modules

The FOS REST API supports 41 Brocade-specific YANG modules.

In the “modules-state/module” list, the server implements the ietf-yang-library module, which identifies all the YANG modules used by the server. For example requests and responses, refer to the Sample Use Cases section of this publication or the RESTCONF RFC (RFC 8040).

The FOS REST API modules supported in Fabric OS 9.1.x are described in the following tables. All support the OPTIONS method.

Table 7: FOS REST API Module Level Version History

Module Name	Description	Supported Methods
brocade-module-version	Allows you to retrieve the module level version information for Brocade Yang modules and submodules.	GET, HEAD, OPTIONS

Table 8: FOS REST API Modules for Brocade Operations Features

Module Name	Description	Supported Methods
brocade-operation-configdownload	Allows you to download a configuration on a switch.	POST, OPTIONS
brocade-operation-configupload	Allows you to upload the configuration on a switch.	POST, OPTIONS
brocade-operation-date	Allows you to configure the system date.	POST, OPTIONS
brocade-operation-device-login-rebalance	Allows you to manage the ports connected to the Fabric Interconnect (FI).	POST, OPTIONS
brocade-operation-fabric	Allows you to initiate a fabric build.	POST, OPTIONS
brocade-operation-extension	Allows you to provide REST RPC support on an extension platform to either clear or default the extension configuration for a switch or a blade. You can also reset the global LAN statistics on a data processor.	POST, OPTIONS
brocade-operation-factory-reset	Allows you to perform a factory reset on a switch.	POST, OPTIONS
brocade-operation-firmwaredownload	Allows you to download and activate firmware from a remote host. You can configure the firmware download to a secondary partition only without rebooting and activation. You can also display, accept, or decline the End User License Agreement (EULA).	POST, OPTIONS
brocade-operation-license	Allows you to install or remove a license on a Brocade switch.	POST, OPTIONS
brocade-operation-lldp	Allows you to clear an LLDP neighbor or reset the LLDP statistics.	POST, OPTIONS
brocade-operation-management-ethernet-interface	Allows you to reset the error statistics of ethernet interface on switch.	POST, OPTIONS
brocade-operation-ntp-clock	Allows you to configure the clock server and key index.	POST, OPTIONS
brocade-operation-pcie-health	Allows you to test the Peripheral Component Interconnect express (PCIe) links between the nontransparent ports of the PCIe switch in the blades and the standby CP.	POST, OPTIONS
brocade-operation-port	Allows you to manage port operations.	POST, OPTIONS
brocade-operation-portchannel	Allows you to reset port channel statistics.	POST, OPTIONS
brocade-operation-port-decommission	Allows you to decommission a port.	POST, OPTIONS
brocade-operation-reboot	Allows you to reboot a switch.	POST, OPTIONS
brocade-operation-restart	Allows you to the IPSec Policy IKE Sessions corresponding to an IPsec policy on an extension platform.	POST, OPTIONS

Module Name	Description	Supported Methods
brocade-operation-sdd-quarantine	Allows you to clear the quarantined state of one or more ports on a device.	POST, OPTIONS
brocade-operation-security-acl-policy	Allows you to configure and manage security-related policies and configurations on a switch.	POST, OPTIONS
brocade-operation-security-authentication-configuration	Allows you to configure and manage authentication-related policies and configurations on a switch.	POST, OPTIONS
brocade-operation-security-authentication-secret	Allows you to configure and manage authentication secrets on a switch.	POST, OPTIONS
brocade-operation-security-certificate	Allows you to import or export a security certificate on a switch.	POST, OPTIONS
brocade-operation-security-fabric-wide-policy-distribute	Allows you to manage security-related policies fabric-wide.	POST, OPTIONS
brocade-operation-security-policy-chassis-distribute	Allows you to manage security-related policies and configurations on a switch chassis.	POST, OPTIONS
brocade-operation-security-policy-distribute	Allows you to manage security-related policies and configurations on switches.	POST, OPTIONS
brocade-operation-security-reset-violation-statistics	Allows you to reset security violation statistics on a switch.	POST, OPTIONS
brocade-operation-security-role-clone	Allows you to clone a security role.	POST, OPTIONS
brocade-operation-show-status	Allows you to retrieve the status of the supportSave operation. You can also view the status of a firmware download operation.	POST, OPTIONS
brocade-operation-supportlink	Allows you to initiate Support Link data collection on the switch, upload data to the Support Link server, and reset the Support Link client to default values.	POST, OPTIONS
brocade-operation-supportsave	Allows you to initiate supportSave and obtain the message ID of the supportSave operation.	POST, OPTIONS
brocade-operation-traffic-optimizer	Allows you to activate the traffic optimizer profile.	POST, OPTIONS
brocade-operation-usb	Allows you to manage the files and folders on the attached USB device.	POST, OPTIONS
brocade-operation-zone	Allows you to perform zone-related operations. You can also perform a zone expunge on zone objects.	POST, OPTIONS

Table 9: FOS REST API Modules for Brocade Fibre Channel Features

Module Name	Description	Supported Methods
brocade-access-gateway	Allows you to retrieve a detailed view of configuration and runtime information of the Access Gateway.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-application-server	Allows you to monitor application-specific services.	GET, HEAD, OPTIONS
brocade-chassis	Allows you to retrieve a detailed view of configuration and runtime information of the chassis.	GET, PATCH, HEAD, OPTIONS
brocade-extension	Allows you to view information about Brocade extension objects as well as configure traffic control lists.	GET, POST, PATCH, HEAD, OPTIONS

Module Name	Description	Supported Methods
brocade-extension-ip-route	Allows you to retrieve or configure IP route information.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-extension-ipsec-policy	Allows you to retrieve or configure the IP security (IPsec) policies on a switch or director.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-extension-tunnel	Allows you to manage Brocade extension tunnels.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-fabric	Allows you to retrieve a list of switches and directors in the fabric.	GET, HEAD, OPTIONS
brocade-fabric-traffic-controller	Allows you to retrieve Fabric Traffic Controller (FTC) diagnostic information for a device.	GET, HEAD, OPTIONS
brocade-fdmi	Allows you to retrieve Fabric Device Management Interface (FDMI) information for the specified switch.	GET, HEAD, OPTIONS
brocade-fibrechannel-configuration	Allows you to configure a switch or director.	GET, PATCH, HEAD, OPTIONS
brocade-fibrechannel-diagnostics	Allows you to perform port diagnostics on a switch or director and retrieve diagnostic port test results.	GET, PATCH, HEAD, OPTIONS
brocade-fibrechannel-logical-switch	Allows you to retrieve information of all logical switches in a operational chassis.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-fibrechannel-routing	Allows you to configure FC routing between two or more fabrics without merging those fabrics.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-fibrechannel-switch	Allows you to retrieve configuration information for a switch or director.	GET, PATCH, HEAD, OPTIONS
brocade-fibrechannel-trunk	Allows you to retrieve a detailed view of all of trunks in the switch in native mode as well as the members of the individual trunks. It can also provide traffic performance and bandwidth information.	GET, POST, DELETE, HEAD, OPTIONS
brocade-ficon	Allows you to manage some FICON Control Unit Port (CUP) as well as providing FICON related information from the Management Server (such as RNID, LIRR, and RLIR data).	GET, PATCH, HEAD, OPTIONS
brocade-firmware	Allows you to manage the firmware version and the integrity of the software image.	GET, PATCH, HEAD, OPTIONS
brocade-fru	Allows you to retrieve a detailed view of configuration and runtime information of the FRUs installed in the chassis which can be especially helpful in monitoring the health of the device.	GET, PATCH, HEAD, OPTIONS
brocade-interface/extension-ip-interface	Allows you to retrieve or configure IP interfaces on the specified switch.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-interface/fibrechannel	Allows you to configure ports and retrieve port configurations.	GET, PATCH, HEAD, OPTIONS

Module Name	Description	Supported Methods
brocade-interface/gigabitethernet	Allows you to retrieve or configure Gigabit Ethernet interfaces or Gigabit Ethernet interface statistics for a switch or director.	GET, PATCH, HEAD, OPTIONS
brocade-interface/portchannel	Allows you to view and configure port channel interfaces on the switch.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-license	Allows you to retrieve a detailed view of the licenses installed on the switch.	GET, HEAD, OPTIONS
brocade-lldp	Allows you to configure and monitor LLDP.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-logging	Allows you to retrieve a detailed view of audit and RASlog message configuration.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-management-ip-interface	Allows you to manage the IP configurations.	GET, PATCH, HEAD, OPTIONS
brocade-maps	Allows you to monitor and configure Monitoring and Alerting Policy Suite features available in the Fabric OS REST API.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-media	Allows you to retrieve a detailed view of the SFP media.	GET, HEAD, OPTIONS
brocade-name-server	Allows you to monitor the operation of one or more instances of Name Server functionality.	GET, HEAD, OPTIONS
brocade-security	Allows you to configure Fabric OS security features for your devices.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-snmp	Allows you to monitor the switch through SNMP queries and trap notifications.	GET, POST, PATCH, HEAD, OPTIONS
brocade-supportlink	Allows you to display the existing SupportLink configuration, enable support link, and configure SupportLink.	GET, PATCH, HEAD, OPTIONS
brocade-time	Allows you to configure the time zone and clock server.	GET, PATCH, HEAD, OPTIONS
brocade-traffic-optimizer	Allows you to view of all switches in the fabric that are operational.	GET, HEAD, OPTIONS
brocade-usb	Allows you to view the files and folders of an attached USB device.	GET, HEAD, OPTIONS
brocade-zone	A data model for configuring zones and retrieving operational zoning information.	GET, POST, PATCH, DELETE, HEAD, OPTIONS

FOS REST API Modules for Operations

This section details the REST RPC support for the FOS REST API operations modules. This section also provides examples for using the FOS REST API operations modules.

brocade-login-response

This module enables you to provides the login response after you log in to a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-login-response.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-login-response

  rpcs:
    +---x login
       +--ro output
          +--ro switch-parameters
             +--ro user-name?                          brocade-security-type:user-config-login-user-name-type
             +--ro user-role?                          brocade-security-type:user-config-role-type
             +--ro chassis-access-role?                brocade-security-type:user-config-role-type
             +--ro home-virtual-fabric?                brocade-security-type:home-virtual-fabric-type
             +--ro virtual-fabric-role-ids
             |  +--ro role-id*   brocade-security-type:virtual-fabric-role-id-type
             +--ro virtual-fabric-enabled?             boolean
             +--ro virtual-fabric-supported?           boolean
             +--ro virtual-fabric-ids
             |  +--ro fabric-id*   fibrechannel:fabric-id-type
             +--ro days-to-user-password-expiration?   uint32
             +--ro firmware-version?                   string
             +--ro model?                              string
```

Examples

This section provides a basic example for using the brocade-login-response module.

Viewing the Login Response

This example uses a POST request to log in to a switch to view the login response details.

Structure

```
POST <base_URI>/login
```

URI

```
POST https://10.10.10.10/rest/login
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <switch-parameters>
        <user-name>admin</user-name>
        <chassis-access-role>admin</chassis-access-role>
        <home-virtual-fabric>128</home-virtual-fabric>
        <virtual-fabric-role-ids>
            <role-id>admin=1-128</role-id>
        </virtual-fabric-role-ids>
        <virtual-fabric-ids>
            <fabric-id>77</fabric-id>
            <fabric-id>128</fabric-id>
        </virtual-fabric-ids>
        <virtual-fabric-supported>true</virtual-fabric-supported>
        <virtual-fabric-enabled>true</virtual-fabric-enabled>
        <firmware-version>v9.1.0</firmware-version>
        <model>173.1</model>
    </switch-parameters>
</Response>
```

brocade-operation-configdownload

This module enables you to download a configuration on a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-configdownload.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Typesfor data type descriptions.

```
module: brocade-operation-configdownload

  rpcs:
    +---x configdownload
       +---w input
       |  +---w configdownload-parameters
       |     +---w config-upload-download-option?   enumeration
       |     +---w port-to-area?                    boolean
       |     +---w vf-id?                           fibrechannel:fabric-id-type
       |     +---w source-vf-id?                    fibrechannel:fabric-id-type
       |     +---w config-input-buffer?             string
       +--ro output
          +--ro show-status
             +--ro message-id?            message-id-type
             +--ro operation?             string
             +--ro status?                string
             +--ro application-name?      string
             +--ro percentage-complete?   uint8
             +--ro firmwaredownload
             |  +--ro message*     string
             |  +--ro eula-text?   string
             +--ro configdownload
                +--ro message*   string
```

Examples

This section provides a basic example for using the brocade-operation-configdownload module.

Downloading a Configuration

This example uses a POST request to download a switch configuration.

Structure

```
POST base_URI/operations/configdownload
```

URI

```
POST https://10.10.10.10/rest/operations/configdownload
```

Request Body

```
<configdownload-parameters>
    <config-upload-download-option>fabric-id</config-upload-download-option>
    <source-vf-id>10</source-vf-id>
```

```
    <config-input-buffer>example-config-payload</config-input-buffer>
</configdownload-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-configupload

This module enables you to restore the configuration on a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-configupload.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-configupload

  rpcs:
    +---x configupload
       +---w input
       |  +---w configupload-parameters
       |     +---w config-upload-download-option?   enumeration
       |     +---w port-to-area?                    boolean
       |     +---w vf-id?                           fibrechannel:fabric-id-type
       +--ro output
          +--ro configupload-operation-status
             +--ro status-message?         string
             +--ro config-output-buffer?   string
```

Examples

This section provides a basic example for using the brocade-operation-configupload module.

Uploading a Configuration to a Switch

This example uses a POST request to upload a configuration to a switch.

Structure

```
POST base_URI/operations/
```

URI

```
POST https://10.10.10.10/rest/operations/
```

Request Body

```
configupload-parameters>
    <config-upload-download-option>switch</config-upload-download-option>
</configupload-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-date

This module enables you to configure the system date.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-operation-date.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-date

  rpcs:
    +---x date
       +---w input
       |  +---w date-operation-parameters
       |     +---w day-of-month    uint16
       |     +---w month           brocade-operation-types:month-type
       |     +---w year            uint16
       |     +---w hour            uint16
       |     +---w minute          uint16
       +--ro output
          +--ro date-operation-status
             +--ro status-message?   string
```

Examples

This section provides a basic example for using the brocade-operation-date module.

Configuring the System Date

This example uses a POST request to configure the date for the system.

Structure

```
POST base_URI/operations/date
```

URI

```
POST https://10.10.10.10/rest/operations/date
```

Request Body

```
<date-operation-parameters>
    <day-of-month>31</day-of-month>
    <month>jul</month>
    <year>1980</year>
    <hour>19</hour>
    <minute>30</minute>
</date-operation-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-device-login-rebalance

This module enables you to manage the ports connected to the Fabric Interconnect (FI).

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-device-login-rebalance.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-device-login-rebalance

  rpcs:
    +---x device-login-rebalance
       +---w input
          +---w device-login-rebalance-parameters
             +---w node-wwn?   fibrechannel:wwn-type
```

Example

This section provides a basic example for using the brocade-operation-device-login-rebalance module.

Rebalancing the Devices Connected to the FI

This section provides a basic example for rebalancing the devices connect to the FI so that no individual port is oversubscribed or undersubscribed.

Structure

```
POST <base_URI>/operations/device-login-rebalance
```

URI

```
POST https://10.10.10.10/rest/operations/device-login-rebalance
```

Request Body

```
<device-login-rebalance-parameters>
    <node-wwn>10:1d:00:ae:ab:a1:17:01</node-wwn>
</device-login-rebalance-parameters>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-operation-diagnostics

This module provides the fibre channel port diagnostics also referred to as the D_Port feature.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-diagnostics.yang-tree.txt file.` See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-diagnostics

  rpcs:
    +---x brocade-diagnostics
       +---w input
          +---w diagnostics-operation-parameters* []
             +---w name                        fibrechannel:slot-port-name-type
             +---w action                      diagnostics-operation-action-type
             +---w frame-count?                yang:counter32
             +---w frame-size?                 uint32
             +---w time?                       string
             +---w payload-pattern
             |  +---w pattern?   pattern-name-type
             |  +---w payload?   fibrechannel:brocade-hex-string-type
             +---w fec-enabled?                boolean
             +---w credit-recovery-enabled?    boolean
```

Examples

This section provides a basic example for using the brocade-login-response module.

Starting Diagnostics on a D_Port

This example uses a POST request to start diagnostics on a D_Port.

Structure

```
POST <base_URI>/operations/brocade-diagnostics
```

URI

```
POST https://10.10.10.10/rest/operations/brocade-diagnostics
```

Request Body

```
<diagnostics-operation-parameters>
    <name>0/25</name>
    <action>start-test</action>
    <frame-count>10</frame-count>
</diagnostics-operation-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “200 OK” status in the first line of the headers.

Stopping Diagnostics on a D_Port

This example uses a POST request to stop diagnostics on a D_Port.

Structure

```
POST <base_URI>/operations/brocade-diagnostics
```

URI

```
POST https://10.10.10.10/rest/operations/brocade-diagnostics
```

Request Body

```
<diagnostics-operation-parameters>
    <name>0/25</name>
    <action>stop-test</action>
</diagnostics-operation-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “200 OK” status in the first line of the headers.

Restarting Diagnostics on a D_Port

This example uses a POST request to restart diagnostics on a D_Port.

Structure

```
POST <base_URI>/operations/brocade-diagnostics
```

URI

```
POST https://10.10.10.10/rest/operations/brocade-diagnostics
```

Request Body

```
<diagnostics-operation-parameters>
    <name>0/25</name>
    <action>restart-test</action>
</diagnostics-operation-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “200 OK” status in the first line of the headers.

brocade-operation-extension

This module provides REST RPC support on an extension platform to either clear or default the extension configuration for a switch or a blade. You can also reset the global LAN statistics on a data processor.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-extension.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-extension

  rpcs:
    +---x extension
       +---w input
       |  +---w extension-operation-parameters {fibrechannel:fibrechannel_extension_platform}?
       |     +---w slot?                          uint16
       |     +---w config-clear?                  boolean
       |     +---w config-default?                boolean
       |     +---w global-lan-statistics-reset?   boolean
       |     +---w dp-id?                         extension:dp-id-type
       +--ro output
          +--ro extension-operation-status {fibrechannel:fibrechannel_extension_platform}?
             +--ro status-message?                string
```

Examples

This section provides basic examples for using the brocade-operation-extension module.

Clearing the Configuration on an Extension Device

This example uses a POST request to clear the configuration on an extension device. This parameter is only configurable when the config-default and global-lan-statistics-reset parameters are set to false.

Note that the config-clear operation may be disruptive. For extension switches, you must reboot the switch after the config-clear operation; otherwise, config operations are blocked. For extension blades, you must power off the blade before you clear the corresponding configuration on the slot.

Structure

POST <*base_URI*>/operations/extension

URI

```
POST https://10.10.10.10/rest/operations/extension
```

Request Body

```
<extension-operation-parameters>
    <slot>0</slot>
    <config-clear>true</config-clear>
</extension-operation-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
    <extension-operation-status>
        <status-message>Operation Succeeded.</status-message>
    </extension-operation-status>
</Response>
```

Resetting an Extension Device to the Default Configuration

This example uses a POST request to reset an extension device to the default configuration.

Note that the config-default operation is a disruptive operation. For extension switches, the extension switch reboots automatically after a successful config-default operation. For extension blades, make sure the blade is in an online state for the config-default operation. You may need to power on and off the blade slot to apply different extension modes correctly.

Structure

POST <*base_URI*>/operations/extension

URI

```
POST https://10.10.10.10/rest/operations/extension
```

Request Body

```
<extension-operation-parameters>
    <slot>0</slot>
    <config-default>true</config-default>
</extension-operation-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
<extension-operation-status>
<status-message>Operation Succeeded.</status-message>
</extension-operation-status>
</Response>
```

Resetting the Global LAN Statistics on a Data Processor

This example uses a POST request to reset the global LAN statistics on a data processor. Depending on the extension platform, the device may have a single DP or dual DP. A single DP supports DP0 only. A dual DP supports both DP0 and DP1. This parameter is only configurable when the global-lan-statistics-reset parameter is set to true.

Structure

POST <*base_URI*>/operations/extension

URI

```
POST https://10.10.10.10/rest/operations/extension
```

Request Body

```
<extension-operation-parameters>
      <slot>0</slot>
      <dp-id>0</dp-id>
       <global-lan-statistics-reset>true</global-lan-statistics-reset>
</extension-operation-parameters>
```

Response Body

```
<?xml version="1.0"?>
    <Response>
     <extension-operation-status>
        <status-message>Operation Succeeded.</status-message>
      </extension-operation-status>
 </Response>
```

brocade-operation-fabric

This module enables you to initiate a fabric build.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-fabric.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-fabric

  rpcs:
    +---x fibrechannel-fabric
       +---w input
          +---w fabric-operation-parameters {fibrechannel:fibrechannel_switch_platform}?
             +---w build-fabric?   boolean
```

Examples

This section provides a basic example for using the brocade-operation-fabric module.

Initiating a Fabric Build

This example uses a POST request to initiate a fabric build.

Structure

POST <*base_URI*>/operations/fibrechannel-fabric

URI

```
POST https://10.10.10.10/rest/operations/fibrechannel-fabric
```

Request Body

```
<fabric-operation-parameters>
    <build-fabric>true</build-fabric>
</fabric-operation-parameters>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

brocade-operation-factory-reset

This module enables you to perform a factory reset on a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-factory-reset.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-factory-reset

  rpcs:
    +---x factory-reset
       +---w input
       |  +---w reset-option-parameters
       |     +---w clear-features
       |     |  +---w feature*   union
       |     +---w keep-features
       |        +---w feature*   brocade-operation-types:keep-feature-type
       |     +---w set-features
       |        +---w feature*   brocade-operation-types:set-feature-type
          +--ro output
          +--ro factory-reset-operation-status
             +--ro status-message?   string
```

Examples

This section provides a basic example for using the brocade-operation-factory-reset module.

Resetting a Switch to the Factory Defaults

This example uses a POST request to reset a switch to the factory defaults.

Structure

```
POST base_URI/operations/factory-reset
```

URI

```
POST https://10.10.10.10/rest/operations/factory-reset
```

Request Body

```
<reset-option-parameters>
    <keep-features>
         <feature>ip</feature>
    </keep-features>
```

Response Body

When the POST operation is successful, the response has a message body similar to the following output, and a “201 Created” status in the headers.

```
<?xml version="1.0"?>
<Response>
```

```
    <factoryreset-operation-status>
        <status-message>Factoryreset operation is completed successfully.</status-message>
    </factoryreset-operation-status>
</Response>
```

Setting a Feature

Clearing the EULA

This example uses a POST request to clear the EULA.

Structure

```
POST base_URI/operations/factory-reset
```

URI

```
POST https://10.10.10.10/rest/operations/factory-reset
```

Request Body

```
<reset-option-parameters>
    <clear-features>
        <feature>eula</feature>
    </clear-features>
</reset-option-parameters>
```

Response Body

When the POST request is successful, the response has a message body similar to the following output and a “200 OK” status in the first line of the headers.

```
<?xml version="1.0"?>
<Response>
    <factoryreset-operation-status>
        <status-message>EULA record cleared successfully.</status-message>
     </factoryreset-operation-status>
</Response>
```

brocade-operation-firmwaredownload

This module enables you to download and activate firmware from a remote host. You can configure the firmware download to a secondary partition only without rebooting and activation. You can also display, accept, or decline the End User License Agreement (EULA).

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-firmwaredownload.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-firmwaredownload

  rpcs:
    +---x firmwaredownload
       +---w input
       |  +---w firmwaredownload-parameters
       |     +---w eula-action?       operation-types:eula-action-type
       |     +---w host?              inet:host
       |     +---w user-name?         brocade-security-type:user-config-user-name-type
       |     +---w password?          string
       |     +---w remote-directory?  string
       |     +---w protocol?          operation-types:protocol
       |     +---w port?              uint32
       |     +---w stage?             boolean
       |     +---w activate?          boolean
       |     +---w firmware-sync?     boolean
       |     +---w clean-install?     boolean
       +--ro output
          +--ro show-status
             +--ro message-id?           message-id-type
             +--ro operation?            string
             +--ro status?               string
             +--ro application-name?     string
             +--ro percentage-complete?  uint8
             +--ro firmwaredownload
             |  +--ro message*     string
             |  +--ro eula-text?   string
             +--ro configdownload
                +--ro message*   string
```

Examples

This section provides basic examples for using the brocade-operation-firmwaredownload module.

Downloading Firmware Using SCP

This example uses a POST request to download firmware using the Secure Copy Protocol (scp). Note that the firmware download process requires that you accept the EULA.

Structure

```
POST <base_URI>/operations/firmwaredownload
```

URI

```
POST https://10.10.10.10/rest/operations/firmwaredownload
```

Request Body

```
<firmwaredownload-parameters>
  <eula-action>accept-eula</eula-action>
  <protocol>scp</protocol>
  <host>10.11.11.11</host>
  <remote-directory>/home/sa023564/GGTEST</remote-directory>
  <user-name>gumbeaux</user-name>
  <password>AZ8byCXdweVfu</password>
  <port>22</port>
</firmwaredownload-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
    <show-status>
        <message-id>20000</message-id>
        <status>queued</status>
        <application-name>Mozilla/5.0 (Windows NT 10.0; Win64; x64</application-name>
        <percentage-complete>0</percentage-complete>
        <operation>firmwaredownload</operation>
        <firmwaredownload>
            <message>Firmwaredownload sanity check in-progress.</message>
        </firmwaredownload>
    </show-status>
</Response>
```

Downloading Firmware Using HTTP

This example uses a POST request to download firmware using the HTTP protocol. Note that the firmware download process requires that you accept the EULA.

Structure

```
POST base_URI/operations/firmwaredownload
```

URI

```
POST https://10.10.10.10/rest/operations/firmwaredownload
```

Request Body

```
<firmwaredownload-parameters>
  <eula-action>accept-eula</eula-action>
  <protocol>http</protocol>
  <host>10.11.11.11</host>
  <remote-directory>/home/sa023564/GGTEST</remote-directory>
  <user-name>gumbeaux</user-name>
```

```
  <password>AZ8byCXdweVfu</password>
  <port>8000</port>
</firmwaredownload-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
    <show-status>
        <message-id>20000</message-id>
        <status>queued</status>
        <application-name>Mozilla/5.0 (Windows NT 10.0; Win64; x64</application-name>
        <percentage-complete>0</percentage-complete>
        <operation>firmwaredownload</operation>
        <firmwaredownload>
            <message>Firmwaredownload sanity check in-progress.</message>
        </firmwaredownload>
    </show-status>
</Response>
```

Viewing the EULA

This example uses a POST request to view the End User License Agreement (EULA).

Structure

```
POST base_URI/operations/firmwaredownload/eula-action/display-eula
```

URI

```
POST https://10.10.10.10/rest/operations/firmwaredownload/eula-action/display-eula
```

Request Body

This is no request body.

Response Body

```
<?xml version="1.0"?>
<Response>
    <text>
        *EULA contents*
    </text>
</Response>
```

Checking the Status of a Firmware Download

This example uses a POST request to check the status of the firmware download.

Structure

```
POST base_URI/operations/show-status/message-id/20000
```

URI

```
POST https://10.10.10.10/rest/operations/show-status/message-id/20000
```

Request Body

This is no request body.

Response Body

```
<?xml version="1.0"?>
<Response>
    <show-status>
        <message-id>20000</message-id>
        <status>queued</status>
        <application-name>Mozilla/5.0 (Windows NT 10.0; Win64; x64</application-name>
        <percentage-complete>0</percentage-complete>
        <operation>firmwaredownload</operation>
        <firmwaredownload>
            <message>Firmwaredownload sanity check in-progress.</message>
        </firmwaredownload>
    </show-status>

</Response>
```

Staging a Firmware Download

This example uses a POST request to stage a firmware download.

Structure

```
POST base_URI/operations/firmwaredownload
```

URI

```
POST https://10.10.10.10/rest/operations/firmwaredownload
```

Request Body

```
<firmwaredownload-parameters>
  <eula-action>accept-eula</eula-action>
  <protocol>scp</protocol>
  <host>10.11.11.11</host>
  <remote-directory>/home/sa023564/GGTEST</remote-directory>
  <user-name>sa023564</user-name>
  <password>TZ8nbWVpbkAx</password>
  <stage>true</stage>
</firmwaredownload-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
    <show-status>
        <message-id>20000</message-id>
        <status>queued</status>
        <application-name>Mozilla/5.0 (Windows NT 10.0; Win64; x64</application-name>
```

```
        <percentage-complete>0</percentage-complete>
        <operation>firmwaredownload</operation>
        <firmwaredownload>
            <message>Firmwaredownload sanity check in-progress.</message>
        </firmwaredownload>
    </show-status>
</Response>
```

Activating Firmware

This example uses a POST request to download the firmware.

Structure

```
POST base_URI/operations/firmwaredownload
```

URI

```
POST https://10.10.10.10/rest/operations/firmwaredownload
```

Request Body

```
<firmwaredownload-parameters>
 <activate>true</activate>
</firmwaredownload-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
    <show-status>
        <message-id>20000</message-id>
        <status>queued</status>
        <application-name>Mozilla/5.0 (Windows NT 10.0; Win64; x64</application-name>
        <percentage-complete>0</percentage-complete>
        <operation>firmwaredownload</operation>
        <firmwaredownload>
            <message>Firmwaredownload sanity check in-progress.</message>
        </firmwaredownload>
    </show-status>
</Response>
```

Downloading Firmware with a Clean Installation

This example uses a POST request to download firmware with a clean installation. Note that the firmware download process requires that you accept the EULA.

Structure

```
POST <base_URI>/operations/firmwaredownload
```

URI

```
POST https://10.10.10.10/rest/operations/firmwaredownload
```

Request Body

```
<firmwaredownload-parameters>
  <eula-action>accept-eula</eula-action>
  <protocol>scp</protocol>
  <host>10.11.11.11</host>
  <remote-directory>/home/jc012345/v9.1.0</remote-directory>
  <user-name>gumbeaux</user-name>
  <password>AZ8byCXdweVfu</password>
  <cleaninstall>true</cleaninstall>
</firmwaredownload-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
  <show-status>
    <message-id>20000</message-id>
    <status>queued</status>
    <application-name>curl/7.19.7 (x86_64-redhat-linux-gnu) li</application-name>

    <percentage-complete>0</percentage-complete>
    <operation>firmwaredownload</operation>
    <firmwaredownload>
      <message>Firmwaredownload sanity check in-progress.</message>
    </firmwaredownload>
  </show-status>
</Response>
```

Initiating Firmware Synchronization

This example uses a POST request to synchronize the firmware on the active CP with the standby CP.

Structure

```
POST <base_URI>/operations/firmwaredownload
```

URI

```
POST https://10.10.10.10/rest/operations/firmwaredownload
```

Request Body

```
<firmwaredownload-parameters>
    <firmware-sync>true</firmware-sync>
</firmwaredownload-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
    <show-status>
        <message-id>20000</message-id>
        <status>queued</status>
        <application-name>Mozilla/5.0 (Windows NT 10.0; Win64; x64</application-name>
        <percentage-complete>0</percentage-complete>
        <operation>firmwaredownload</operation>
```

```
        <firmwaredownload>
            <message>Firmwaredownload sanity check in-progress.</message>
        </firmwaredownload>
    </show-status>
</Response>
```

brocade-operation-license

This module enables you to install or remove a license on a Brocade switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-license.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-license

  rpcs:
    +---x license
       +---w input
       |  +---w license-parameters
       |     +---w name?               string
       |     +---w action?             enumeration
       |     +---w license-payload?    string
       |     +---w usb-file-path?      string
       |     +---w host?               inet:host
       |     +---w user-name?          string
       |     +---w password?           string
       |     +---w remote-directory?   string
       |     +---w protocol?           operation-types:protocol
       |     +---w port?               uint32
       |     +---w remote-file?        string
       +--ro output
          +--ro license-operation-status
             +--ro status-message?   string
```

Examples

This sections provides a basic example for using the brocade-operation-license module.

Installing a License Using an XML File

This example uses a POST request to install a license using an XML file on a Brocade switch.

NOTE
You must use this method to install a license on a Gen 7 platform.

Structure

```
POST <base_URI>/operations/license
```

URI

```
POST https://10.10.10.10/rest/operations/license
```

Request Body

```
<license-parameters>
    <action>install</action>
    <host>10.10.10.25</host>
    <user-name>adminaccount</user-name>
```

```
    <password>xxxxxxxxxx</password>
    <remote-directory>/var/ftp/pub/private/ab123456/lc/lc_10_20190503.xml</remote-directory>
    <protocol>scp</protocol>
</license-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
    <license-parameters>
        <status-message>License Install success</status-message>
    </license-parameters>
</Response>
```

Installing a License Using a License Key

This example uses a POST request to install a license key on a Brocade switch.

NOTE
You must use this method to install a license on a Gen 6 platform.

Structure

```
POST <base_URI>/operations/license
```

URI

```
POST https://10.10.10.10/rest/operations/license
```

Request Body

```
<license-parameters>
    <action>install</action>
    <name>RRmtMRGYg4MZ9EPHaWaGgBCHBtCEaDCLBAQYH</name>
</license-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
    <license-parameters>
        <status-message>For license change to take effect, please reboot the switch now...</status-message>
    </license-parameters>
</Response>
```

Removing a License from a Gen 7 Platform

This example uses a POST request to remove a license from a Gen 7 platform.

Structure

```
POST <base_URI>/operations/license
```

URI

```
POST https://10.10.10.10/rest/operations/license
```

Request Body

```
<license-parameters>
    <action>remove</action>
    <name>FOS-01-0-03-11201003</name>
</license-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
    <license-parameters>
        <status-message>License Remove success</status-message>
    </license-parameters>
</Response>
```

Removing a License from a Gen 6 Platform

This example uses a POST request to remove a license from a Brocade switch.

Structure

```
POST <base_URI>/operations/license
```

URI

```
POST https://10.10.10.10/rest/operations/license
```

Request Body

```
<license-parameters>
    <action>remove</action>
    <name>RRmtMRGYg4MZ9EPHaWaGgBCHBtCEaDCLBAQYH</name>
</license-parameters>
```

Response Body

```
<?xml version="1.0"?>
<Response>
    <license-parameters>
        <status-message>For license change to take effect, please reboot the switch now...</status-message>
    </license-parameters>
</Response>
```

brocade-operation-lldp

This module enables you to clear an LLDP neighbor or reset the LLDP statistics.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-lldp.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-lldp

  rpcs:
    +---x lldp
       +---w input
          +---w lldp-operations
             +---w slot-port?            fibrechannel:slot-port-name-type
             +---w cp-name?              fibrechannel:cp-name-type
             +---w physical-interface?   fibrechannel:physical-interface-type
             +---w action?               enumeration
```

Examples

This sections provides a basic example for using the brocade-operation-lldp module.

Clearing an LLDP Neighbor

This example uses a POST request to clear the LLDP neighbor details for a specified port.

Structure

```
POST base_URI/operations/lldp
```

URI

```
POST https://10.10.10.10/rest/operations/lldp
```

Request Body

```
<lldp-operations>
    <slot-port>0/3</slot-port>
    <action>clear-neighbor</action>
</lldp-operations>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Resetting LLDP Statistics

This example uses a POST request to reset the LLDP statistics for a specified port.

Structure

```
POST base_URI/operations/lldp
```

URI

```
POST https://10.10.10.10/rest/operations/lldp
```

Request Body

```
<lldp-operations>
    <slot-port>0/3</slot-port>
    <action>reset-statistics</action>
</lldp-operations>
```

Response Body

When the statistics are successfully reset, the response contains an empty message body and a "204 No Content" status in the headers.

brocade-operation-management-ethernet-interface

This modules enables you to reset the error statistics of an Ethernet interface on a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-management-ethernet-interface.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-management-ethernet-interface

  rpcs:
    +---x management-ethernet-interface
       +---w input
          +---w ethernet-interface-parameters
             +---w physical-interface?   fibrechannel:physical-interface-type
             +---w cp-name?              fibrechannel:cp-name-type
             +---w action?               enumeration
```

Examples

This sections provides a basic example for using the brocade-operation-management-ethernet-interface module.

Resetting the Error Statistics on an Ethernet Interface

This example uses a POST request to reset the error statistics on an Ethernet interface.

Structure

```
POST base_URI/operations/management-ethernet-interface
```

URI

```
POST https://10.10.10.10/rest/operations/management-ethernet-interface
```

Request Body

```
<ethernet-interface-parameters>
    <cp-name>cp0</cp-name>
    <physical-interface>eth0</physical-interface>
    <action>reset-error</action>
</ethernet-interface-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-ntp-clock-server

This module enables you to configure the clock server and key index.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-ntp-clock-server.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-ntp-clock-server

  rpcs:
    +---x ntp-clock-server
       +---w input
          +---w clock-server-parameters* []
             +---w action?   brocade-time-type:ts-operation-type
             +---w server?   ts-ntp-type
             +---w index?    int32
```

Examples

This section provides a basic example for using the brocade-operation-time module.

Configuring the Clock Server and Key Index

This example uses a POST request to configure the clock server and key index.

Structure

```
POST base_URI/operations/time
```

URI

```
POST https://10.10.10.10/rest/operations/time
```

Request Body

```
<clock-server-parameters>
    <action>create</action>
    <server>10.10.10.10</server>
    <index>11</index>
</clock-server-parameters>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status in the header.

Editing the Clock Server and Key Index

This example uses a POST request to edit the clock server and key index.

Structure

```
POST base_URI/operations/time
```

URI

```
POST https://10.10.10.10/rest/operations/time
```

Request Body

```
<clock-server-parameters>
    <action>modify</action>
    <server>10.10.10.10</server>
    <index>5</index>
</clock-server-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-pcie-health

This module enables you to test the Peripheral Component Interconnect express (PCIe) links between the non-transparent ports of the PCIe switch in the blades and the standby CP.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-pcie-health.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-pcie-health

  rpcs:
    +---x pcie-health-test
       +---w input
       |  +---w slot-test {fibrechannel:fibrechannel_chassis_platform}?
       |     +---w slot-id?      uint8
       +--ro output
          +--ro slot-test* []
             +--ro slot-id?      uint8
             +--ro status?       string
```

Examples

This sections provides a basic example for using the brocade-operation-pcie-health module.

Testing the PCIe Links

This example uses a POST request to test the Peripheral Component Interconnect express (PCIe) links between the non-transparent ports of the PCIe switch in the blades and the standby CP.

Structure

```
POST base_URI/operations/pcie-health-test
```

URI

```
POST https://10.10.10.10/rest/operations/pcie-health-test
```

Request Body

The request body is empty.

```
<slot-test>
  <slot-id>255</slot-id>
</slot-test>
```

Response Body

```
<?xml version="1.0"?>
<Response>
  <slot-test>
    <slot-id>4</slot-id>
    <status>SUCCESS.</status>
  </slot-test>
  <slot-test>
    <slot-id>7</slot-id>
    <status>SUCCESS.</status>
  </slot-test>
  <slot-test>
    <slot-id>8</slot-id>
    <status>SUCCESS.</status>
  </slot-test>
  <slot-test>
    <slot-id>12</slot-id>
    <status>SUCCESS.</status>
  </slot-test>
</Response>
```

brocade-operation-port

This modules enables you to manage port operations.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-port.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-port

  rpcs:
    +---x port
       +---w input
          +---w port-operation-parameters* []
             +---w name?                fibrechannel:slot-port-name-type
             +---w operation-type?      port-operation-type
             +---w user-port-address?   fibrechannel:area-address-type
             +---w auto-bind?           boolean
```

Examples

This sections provides basic examples for using the brocade-operation-port module.

Binding a Port Address

This example uses a POST request to bind a port address.

Structure

```
POST base_URI/operations/port
```

URI

```
POST https://10.10.10.10/rest/operations/port
```

Request Body

```
<port-operation-parameters>
    <name>5/18</name>
    <operation-type>port-address-bind</operation-type>
    <user-port-address>0x1a00</user-port-address>
    <auto-bind>true</auto-bind>
</port-operation-parameters>
```

Response Body

When the POST request is successful, the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

Unbinding a Port Address

This example uses a POST request to unbind a port address.

Structure

```
POST base_URI/operations/port
```

URI

```
POST https://10.10.10.10/rest/operations/port
```

Request Body

```
<port-operation-parameters>
    <name>5/18</name>
    <operation-type>port-address-unbind</operation-type>
    <user-port-address>0x1a00</user-port-address>
<port-operation-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

Resetting the FC Port Statistics

This example uses a POST request to reset statistics for an FC port.

Structure

```
POST base_URI/operations/port
```

URI

```
POST https://10.10.10.10/rest/operations/port
```

Request Body

```
<port-operation-parameters>
    <name>5/18</name>
    <operation-type>port-reset-statistics</operation-type>
</port-operation-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “200 OK” status in the first line of the headers.

Resetting the GE Port Statistics

This example uses a POST request toreset statistics for an GE port.

Structure

```
POST base_URI/operations/port
```

URI

```
POST https://10.10.10.10/rest/operations/port
```

Request Body

```
<port-operation-parameters>
```

```
    <name>1/10</name>
    <operation-type>extension-ge-port-reset-statistics</operation-type>
</port-operation-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “200 OK” status in the first line of the headers.

brocade-operation-portchannel

This module enables you to reset port channel statistics.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-portchannel.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-portchannel

  rpcs:
    +---x portchannel
       +---w input
          +---w portchannel-operation-parameters {fibrechannel:fibrechannel_chassis_platform
                or fibrechannel:fibrechannel_extension_platform}?
             +---w name?     brocade-interface-types:portchannel-interface-name-type
             +---w action?   enumeration
```

Examples

This sections provides a basic example for using the brocade-operation-portchannel module.

Resetting the Port Channel Statistics

This example uses a POST request to reset port channel statistics.

Structure

```
POST base_URI/operations/portchannel
```

URI

```
POST https://10.10.10.10/rest/operations/portchannel
```

Request Body

```
<portchannel-operation-parameters>
    <name>1</name>
    <action>reset-statistics</action>
</portchannel-operation-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-port-decommission

This module enables you to decommission a port.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-port-decommission.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-port-decommission

  rpcs:
    +---x port-decommission
       +---w input
       |  +---w port-decommission-parameters* []
       |     +---w slot-port?                fibrechannel:slot-port-name-type
       |     +---w port-decommission-type?   port-decommission-operation-type
       +--ro output
          +--ro show-status
             +--ro message-id?            message-id-type
             +--ro operation?             string
             +--ro status?                string
             +--ro application-name?      string
             +--ro percentage-complete?   uint8
             +--ro firmwaredownload
             |  +--ro message*     string
             |  +--ro eula-text?   string
             +--ro configdownload
                +--ro message*   string
```

Examples

This section provides basic examples for using the brocade-operation-port-decommission module.

Decommissioning a Port

This example uses a POST request to decommissions a port.

Structure

```
POST base_URI/operations/port-decommission
```

URI

```
POST https://10.10.10.10/rest/operations/port-decommission
```

Request Body

```
<port-decommission-parameters>
    <slot-port>1/18</slot-port>
    <port-decommission-type>port</port-decommission-type>
</port-decommission-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

Decommissioning a QSFP module

This example uses a POST request to decommissions a QSFP module.

Structure

```
POST base_URI/operations/port-decommission
```

URI

```
POST https://10.10.10.10/rest/operations/port-decommission
```

Request Body

```
<port-decommission-parameters>
    <slot-port>1/18</slot-port>
    <port-decommission-type>qsfp-port</port-decommission-type>
</port-decommission-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-reboot

This module enables you to reboot a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-reboot a switch.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-reboot

  rpcs:
    +---x reboot
       +---w input
       |  +---w reboot-operation-parameters
       |     +---w action?   brocade-operation-types:reboot-types
       +--ro output
          +--ro reboot-operation-status
             +--ro status-message?   string
```

Examples

This section provides a basic example for using the brocade-operation-reboot module.

Rebooting a Switch

This example uses a POST request to perform a cold reboot on a switch.

Structure

```
POST base_URI/operations/reboot
```

URI

```
POST https://10.10.10.10/rest/operations/reboot
```

Request Body

The request body is empty.

```
<reboot-operation-parameters>
    <action>reboot</action>
</reboot-operation-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-restart

This module contains a collection of YANG definitions for managing restart operations.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-restart.yang-tree.txt file.` See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-restart

  rpcs:
    +---x restart
       +---w input
          +---w restart-operation-parameters* []
             +---w action?                          restart-operation-action-type
             +---w extension-ipsec-policy-name?     extension:extension-name-type
```

Examples

This section provides a basic example for using the brocade-operation-restart module.

Restarting the IPSec Policy IKE Sessions

This example uses a POST request to restart the IPSec Policy IKE Sessions corresponding to an IPsec policy on an extension platform.

Structure

```
POST <base_URI>/operations/restart
```

URI

```
POST https://10.10.10.10/operations/restart
```

Request Body

```
{
    "restart-operation-parameters": {
        "operation" : "restart_ike_sessions",
        "extension-ipsec-policy-name" : "test"
    }
}
```

Response Body

When the POST request is successful, the response contains an empty message body and a “200 OK” status in the first line of the headers.

brocade-operation-sdd-quarantine

This module contains a collection of YANG definitions for SDD Quarantine operations.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-sdd-quarantine.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-sdd-quarantine

  rpcs:
    +---x sdd-quarantine
       +---w input
          +---w sdd-quarantine-parameters
             +---w action?       enumeration
             +---w port-index?   fibrechannel:port-number-type
             +---w force?        boolean
```

Examples

This section provides basic examples for using the brocade-operation-sdd-quarantine module.

Clearing the Quarantined State of a Local Switch Port

This example uses a POST request to clear the quarantined state of port 9.

Structure

```
POST base_URI/operations/quarantine-clear/port-index/
```

URI

```
POST https://10.10.10.10/rest/operations/quarantine-clear/port-index/
```

Request Body

```
<sdd-quarantine-parameters>
    <action>clear</action>
    <port-index>9</port-index>
    <force>true</force>
</sdd-quarantine-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

Clearing the Quarantined State of all Local Switch Ports

This example uses a POST request to clear the quarantined state of all ports on the device.

Structure

```
POST base_URI/operations/quarantine-clear/
```

URI

```
POST https://10.10.10.10/rest/operations/quarantine-clear/
```

Request Body

```
<sdd-quarantine-parameters>
    <action>clear-all</action>
</sdd-quarantine-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-security-acl-policy

This module enables you to configure and manage security-related policies and configurations on a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-security-acl-policy.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-security-acl-policy

  rpcs:
    +---x security-acl-policy
       +---w input
          +---w security-policy-parameters* []
             +---w policy-name?      brocade-security-type:security-policy-name-type
             +---w members
             |  +---w member*   brocade-security-type:security-input-type
             +---w switch-ports
             |  +---w port*   union
             +---w include-device?   boolean
             +---w action?           brocade-security-type:security-policy-action-type
             +---w fcs-position?     uint16
```

Examples

This section provides a basic example for using the brocade-operation-security-acl-policy module.

Configuring the Security ACL Policy

This example uses a POST request to configure the security ACL policy.

Structure

```
POST base_URI/operations/security-acl-policy
```

URI

```
POST https://10.10.10.10/rest/operations/security-acl-policy
```

Request Body

```
<security-acl-parameters>
    <policy-name> SCC_POLICY</policy-name>
    <members>
        <member>12:24:45:10:0a:67:00:40</member>
        <member>12:24:45:10:0a:67:00:50</member>
     <members>
     <action>create</action>
</security-acl-parameters>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-operation-security-authentication-configuration

This module enables you to configure and manage authentication-related policies and configurations on a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-security-authentication-configuration.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-security-authentication-configuration

  rpcs:
    +---x security-authentication-configuration
       +---w input
          +---w authentication-configuration-parameters
             +---w authentication-protocols
             |  +---w protocol*   brocade-security-type:authentication-protocol-type
             +---w authentication-hashes
             |  +---w hash*   brocade-security-type:authentication-hash-type
             +---w dh-group?                          brocade-security-type:dh-group-type
             +---w switch-authentication-policy?      brocade-security-type:authentication-policy-mode-type
             +---w device-authentication-policy?      brocade-security-type:authentication-policy-mode-type
             +---w authentication-init-slot-ports {fibrechannel:fibrechannel_switch_platform}?
                +---w port*   brocade-security-type:authentication-initiation-parameter-type
```

Examples

This section provides a basic example for using the brocade-operation-security-authentication-configuration module.

Configuring the Authentication Policy

This example uses a POST request to configure the authentication policy.

Structure

```
POST base_URI/operations/security-authentication-configuration
```

URI

```
POST https://10.10.10.10/rest/operations/security-authentication-configuration
```

Request Body

```
<authentication-configuration-parameters>
    <authentication-protocols>
        <protocol>dh-chap</protocol>
    <authentication-protocols>
    <authentication-hashes>
        <hash>sha256</hash>
    </authentication-hashes>
    <dh-group>dh-group-all</dh-group>
    <switch-authentication-policy>on</switch-authentication-policy>
    <authentication-init-slot-ports>
```

```
        <port>18</ports>
    </authentication-init-slot-ports>
</authentication-configuration-parameters>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-operation-security-authentication-secret

This module enables you to configure and manage authentication secrets on a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-security-authentication-secret.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-security-authentication-secret

  rpcs:
    +---x security-authentication-secret
       +---w input
       |  +---w authentication-secret-parameters* []
       |     +---w member?         brocade-security-type:security-input-type
       |     +---w peer-secret?    brocade-security-type:authentication-secret-type
       |     +---w local-secret?   brocade-security-type:authentication-secret-type
       |     +---w action?         brocade-security-type:authentication-secret-action-type
       +--ro output
          +--ro authentication-secret-operation-status* []
             +--ro status-message?   string
```

brocade-operation-security-authentication-secret Examples

This section provides a basic example for using the brocade-operation-security-authentication-secret module.

Configuring the Shared Secret for DH_CHAP Authentication

This example uses a POST request to configure the shared secret for DH_CHAP authentication.

Structure

```
POST base_URI/operations/security-authentication-secret
```

URI

```
POST https://10.10.10.10/rest/operations/security-authentication-secret
```

Request Body

```
<authentication-secret-parameters>
    <member>10:00:00:05:33:e7:b3:60</member>
    <peer-secret>peer_secret</peer-secret>
    <local-secret>local_secret</local-secret>
    <action>set</action>
</authentication-secret-parameters>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-operation-security-certificate

This module enables you to import or export a security certificate on a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-security-certificate.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-security-certificate

  rpcs:
    +---x security-certificate
       +---w input
          +---w security-certificate-parameters
             +---w certificate-entity?    brocade-security-type:certificate-entity-type
             +---w certificate-type?      brocade-security-type:certificate-application-type
             +---w certificate-name?      string
             +---w ca-certificate?        string
             +---w keypair-tag?           string
             +---w action?                brocade-security-type:seccertmgmt-operation-type
             +---w certificate-hexdump?   brocade-security-type:certificate-hexdump-type
```

Examples

This section provides a basic example for using the brocade-operation-security-certificate module.

Installing a Switch Certificate

This example uses a POST request to install a switch certificate.

Structure

```
POST base_URI/operations/security-certificate
```

URI

```
POST https://10.10.10.10/rest/operations/security-certificate
```

Request Body

```
<security-certificate-parameters>
    <certificate-entity>CA</certificate-entity>
    <certificate-type>https</certificate-type>
    <certificate-name>PADL12R2SHA256CA.pem</certificate-name>
    <action>import</action>
    <certificate-hexdump> </certificate-hexdump>
</security-certificate-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-security-fabric-wide-policy-distribute

This module enables you to manage security-related policies fabric-wide.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-security-fabric-wide-policy-distribute.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-security-fabric-wide-policy-distribute

  rpcs:
    +---x security-fabric-wide-policy-distribute
       +---w input
          +---w distribute-parameters
             +---w policy-types
                +---w type*   brocade-security-type:fabric-wide-consistency-policy-type
```

Examples

This section provides a basic example for using the brocade-operation-security-fabric-wide-policy-distribute module.

Configuring the Fabric-Wide Consistency Policy

This example uses a POST request to configuring the fabric-wide consistency policy.

Structure

```
POST base_URI/operations/security-fabric-wide-policy-distribute
```

URI

```
POST https://10.10.10.10/rest/operations/security-fabric-wide-policy-distribute
```

Request Body

```
<distribute-parameters>
    <policy-types>
        <type>strict-scc</type>
    </policy-types>
</distribute-parameters>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-operation-security-policy-chassis-distribute

This module enables you to manage security-related policies and configurations on a switch chassis.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-security-policy-chassis-distribute.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-security-policy-chassis-distribute

  rpcs:
    +---x security-policy-chassis-distribute
       +---w input
       |  +---w policy-chassis-distribute-parameters {fibrechannel:fibrechannel_switch_platform}?
       |     +---w policy-types
       |     |  +---w type*   brocade-security-type:chassis-distribute-policy-type
       |     +---w destination-domains {fibrechannel:fibrechannel_non_virtual_fabric_platform}?
       |     |  +---w domain-id*   brocade-security-type:chassis-distribute-domain-type
       |     +---w vf-id?                 fibrechannel:fabric-id-type
 {fibrechannel:fibrechannel_virtual_fabric_platform}?
       +--ro output
          +--ro chassis-distribute-operation-status* []
             +--ro switch-wwn?             fibrechannel:wwn-type
             +--ro domain-id?              fibrechannel:domain-id-type
             +--ro switch-name?            string
             +--ro vf-id?                  fibrechannel:fabric-id-type
 {fibrechannel:fibrechannel_virtual_fabric_platform}?
             +--ro distribute-supported?   boolean
```

Examples

This section provides a basic example for using the brocade-operation-security-policy-chassis-distribute module.

Configuring the IP Filter Policies on a Chassis

This example uses a POST request to configuring the chassis wide distribution of IP filter policies.

Structure

```
POST base_URI/operations/security-policy-chassis-distribute
```

URI

```
POST https://10.10.10.10/rest/operations/security-policy-chassis-distribute
```

Request Body

```
<policy-chassis-distribute-parameters>
    <policy-types>
        <type>ipfilter</type>
    </policy-types>
    <destination-domains>
```

```
        <domain-id>11</domain-id>
    </destination-domains>
</policy-chassis-distribute-parameters>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-operation-security-policy-distribute

This module enables you to manage security-related policies and configurations on switches.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-security-policy-distribute.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-security-policy-distribute

  rpcs:
    +---x security-policy-distribute
       +---w input
       |  +---w policy-distribute-parameters
       |     +---w distribute-policy-types
       |     |  +---w type*   brocade-security-type:distribute-policy-type
       |     +---w destination-names
       |        +---w name*   brocade-security-type:security-input-type
       +--ro output
          +--ro distribute-operation-status
             +--ro status-message?   string
```

Examples

This section provides a basic example for using the brocade-operation-security-policy-distribute module.

Configuring the Security Policies to be Distributed

This example uses a POST request to configuring the security policies to be distributed.

Structure

```
POST base_URI/operations/security-policy-distribute
```

URI

```
POST https://10.10.10.10/rest/operations/security-policy-distribute
```

Request Body

```
<policy-distribute-parameters>
    <distribute-policy-types>
        <type>scc</type>
    </distribute-policy-types>
    <destination-names>
        <name>11</name>
    </destination-names>
</policy-distribute-parameters>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-operation-security-reset-violation-statistics

This module enables you to reset security violation statistics on a switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-security-reset-violation-statistics.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-security-reset-violation-statistics

  rpcs:
    +---x security-reset-violation-statistics
       +---w input
          +---w violation-statistics
             +---w domain-id?         fibrechannel:domain-id-type
             +---w statistics-name?   brocade-security-type:security-violation-statistics-type
```

Examples

This section provides a basic example for using the brocade-operation-security-reset-violation-statistics module.

Resetting the Security Violation Statistics on a Switch

This example uses a POST request to reset the login violation statistics on a switch.

Structure

```
POST base_URI/operations/security-reset-violation-statistics
```

URI

```
POST https://10.10.10.10/rest/operations/security-reset-violation-statistics
```

Request Body

```
<domain-id>43</domain-id>
<statistics-name>login</statistics-name>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-security-role-clone

This module enables you to clone a security role.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-security-role-config.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-security-role-clone

  rpcs:
    +---x security-role-clone
       +---w input
          +---w security-role-clone-parameters
             +---w name           brocade-security-type:user-config-role-type
             +---w source-name    brocade-security-type:user-config-role-type
```

Examples

This section provides a basic example for using the brocade-operation-security-role-config module.

Copying a Security Role

This example uses a POST request to create a new security role based on an existing security role.

Structure

```
POST base_URI/operations/security-role-clone
```

URI

```
POST https://10.10.10.10/rest/operations/security-role-clone
```

Request Body

```
<name>NewSecurityRole</name>
<source-name>OldSecurityRole</source-name>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-operation-show-status

This module enables you to obtain the status of any REST operation.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-show-status.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-show-status
   +---x show-status
      +---w input
      | +---w message-id                operation:message-id-type
      +--ro output
         +--ro show-status
            +--ro message-id?            message-id-type
            +--ro operation?             string
            +--ro status?                string
            +--ro application-name?      string
            +--ro percentage-complete?   uint8
            +--ro firmwaredownload?
              +--ro message*             string
              +--ro eula-text?           string
```

brocade-operation-show-status Examples

This section provides basic examples for using the brocade-operation-show-status module.

Viewing the supportSave Status

The following example uses the POST request to obtain the status of a supportSave operation.

Structure

`POST <base_URI>/operations/show-status/message-id/message-id` (where *message-id* is the message ID generated when you initiated the supportSave operation)

URI

```
POST https://10.10.10.10/rest/operations/show-status/message-id/99767053
```

Request Body

There is no request body.

Response Body

```
<?xml version="1.0"?>
<Response>
  <show-status>
    <message-id>99767053</message-id>
    <status>progress</status>
    <application-name>PostmanRuntime/6.1.6</application-name>
```

```
    <perentage-complete>35</perentage-complete>
    <operation>supportsave</operation>
  </show-status>
</Response>
```

Viewing the Firmware Download Status

The following example uses the POST request to obtain the status of a firmware download operation.

Structure

`POST <base_URI>/operations/show-status/message-id/message-id` (where *message-id* is the message ID generated when you initiated the firmwaredownload operation)

URI

```
POST https://10.10.10.10/rest/operations/show-status/message-id/20000
```

Request Body

There is no request body.

Response Body

```
<?xml version="1.0"?>
<Response>
    <show-status>
        <message-id>20000</message-id>
        <status>queued</status>
        <application-name>Mozilla/5.0 (Windows NT 10.0; Win64; x64</application-name>
        <percentage-complete>0</percentage-complete>
        <operation>firmwaredownload</operation>
        <firmwaredownload>
            <message>Firmwaredownload sanity check in-progress.</message>
        </firmwaredownload>
    </show-status>
    </Response>
```

brocade-operation-supportlink

This module enables you to initiate Support Link data collection on the switch, upload data to the Support Link server, and reset the Support Link client to default values.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-supportlink.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-supportlink

  rpcs:
    +---x supportlink
       +---w input
          +---w supportlink
             +---w action?   enumeration
```

Examples

This section provides basic examples for using the brocade-operation-supportlink module.

Resetting the Support Link Values to Defaults

This example uses a POST request to reset the Support Link values to defaults on the specified switch.

Structure

```
POST base_URI/operations/supportlink
```

URI

```
POST https://10.10.10.10/rest/operations/supportlink
```

Request Body

```
<supportlink>
  <action>display</action>
</supportlink>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Collecting Support Link Data

This example uses a POST request to collect Support Link data from the specified switch.

Structure

```
POST base_URI/operations/supportlink/action/collect
```

URI

```
POST https://10.10.10.10/rest/operations/supportlink/action/collect
```

Request Body

```
<supportlink>
  <action>collect</action>
</supportlink>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Uploading Support Link Data

This example uses a POST request to upload Support Link data from the specified switch.

Structure

```
POST base_URI/operations/supportlink
```

URI

```
POST https://10.10.10.10/rest/operations/supportlink
```

Request Body

```
<supportlink>
  <action>send</action>
</supportlink>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-operation-supportsave

This module enables you to initiate supportSave and obtain the message ID of the supportSave operation.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-operation-supportsave.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-supportsave

  rpcs:
    +---x supportsave
       +---w input
       |  +---w connection
       |     +---w host?                 inet:host
       |     +---w user-name?            string
       |     +---w password?             string
       |     +---w remote-directory?     string
       |     +---w protocol?             string
       |     +---w serial-mode?          boolean
       |     +---w port?                 uint32
       +--ro output
          +--ro show-status
             +--ro message-id?           message-id-type
             +--ro operation?            string
             +--ro status?               string
             +--ro application-name?     string
             +--ro percentage-complete?  uint8
             +--ro firmwaredownload
                +--ro message*           string
                +--ro eula-text?         string
             +--ro configdownload
                +--ro message*           string
```

Examples

The following example uses the POST request to initiate supportSave on a switch.

Initiating supportSave

Structure

```
POST <base_URI>/operations/supportsave
```

URI

```
POST https://10.10.10.10/rest/operations/supportsave
```

Request Body

```
<connection>
  <protocol>scp</protocol>
  <host>10.10.10.10</host>
  <port>123</port>
  <remote-directory>/supportsave/test</remote-directory>
  <user-name>gumbeaux</user-name>
  <password>aZByc3xdwdV=</password>
</connection>
```

Response Body

```
<?xml version="1.0"?>
<Response>
  <show-status>
    <message-id>99767053</message-id>
    <status>queued</status>
    <application-name>PostmanRuntime/6.1.6</application-name>
    <operation>supportsave</operation>
    <percentage-complete>0</percentage-complete>
    </show-status>
</Response>
```

To see the status of the supportSave request, see Viewing the supportSave Status.

brocade-operation-traffic-optimizer

This module enables you to activate the Traffic Optimizer profile.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-traffic-optimizer.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-traffic-optimizer

  rpcs:
    +---x traffic-optimizer
       +---w input
          +---w traffic-optimizer-parameters
          {fibrechannel:fibrechannel_traffic_optimizer_platform}?
             +---w action?   enumeration
             +---w name?     string
             +---w delay?    uint16
             +---w action?   enumeration
             +---w speeds
                +---w speed*    fibrechannel:protocol-speed-type
```

Examples

This section provides a basic example for using the brocade-operation-traffic-optimizer module.

Activating the Traffic Optimizer Profile

This example uses a POST request to activate the traffic optimizer Profile_Version1.

Structure

```
POST base_URI/operations/traffic-optimizer
```

URI

```
POST https://10.10.10.10/rest/operations/traffic-optimizer
```

Request Body

```
<traffic-optimizer-parameters>
   <name>Profile_Version1</name>
   <action>activate-all-vf</action>
</traffic-optimizer-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

Scheduling Activation of the Traffic Optimizer Profile

This example uses a POST request to activate traffic optimizer Profile_Version2 at a specific time.

Structure

```
POST base_URI/operations/traffic-optimizer
```

URI

```
POST https://10.10.10.10/rest/operations/traffic-optimizer
```

Request Body

```
<traffic-optimizer-parameters>
  <name>Profile_Version2</name>
  <delay>60</delay>
  <action>schedule</action>
</traffic-optimizer-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

Stopping the Activation of the Traffic Optimizer Profile

This example uses a POST request to stop the activation traffic optimizer Profile_Version2.

Structure

```
POST base_URI/operations/traffic-optimizer
```

URI

```
POST https://10.10.10.10/rest/operations/traffic-optimizer
```

Request Body

```
<traffic-optimizer-parameters>
    <name>Profile_Version2</name>
    <action>abort-schedule</action>
</traffic-optimizer-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

Adding a Member to a Custom Profile

This example uses a POST request to add a member (trafopt_custom) to a traffic optimizer profile.

Structure

```
POST base_URI/operations/traffic-optimizer
```

URI

```
POST https://10.10.10.10/rest/operations/traffic-optimizer
```

Request Body

```
<traffic-optimizer-parameters>
    <action>add-config-member</action>
    <name>trafopt_custom</name>
```

```
    <protocol>nvme</protocol>
    <speeds>
      <speed>64</speed>
    </speeds>
</traffic-optimizer-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

Removing a Member from a Custom Profile

This example uses a POST request to remove a member (trafopt_custom) from a traffic optimizer profile.

Structure

```
POST base_URI/operations/traffic-optimizer
```

URI

```
POST https://10.10.10.10/rest/operations/traffic-optimizer
```

Request Body

```
<traffic-optimizer-parameters>
    <action>remove-config-member</action>
    <name>trafopt_custom</name>
    <protocol>nvme</protocol>
    <speeds>
      <speed>64</speed>
    </speeds>
</traffic-optimizer-parameters>
```

Response Body

When the POST request is successful, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-usb-delete-file

This module enables you to manage the files and folders on the attached USB device.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-usb-delete-file.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-usb-delete-file

  rpcs:
    +---x usb-delete-file
       +---w input
          +---w usb-operation-parameters
             +---w path?   operation-types:usb-path-type
             +---w name?   operation-types:usb-filename-type
```

Examples

This section provides a basic example for using the brocade-operation-usb module.

Deleting a File from the USB

This example uses a POST request to delete a file from the USB.

Structure

```
POST base_URI/operations/usb-delete-file/
```

URI

```
POST https://10.10.10.10/rest/operations/usb-delete-file/
```

Request Body

```
<usb-operation-parameters>
    <path>/usb/usbstorage/</path>
    <name>config_new.txt</name>
</usb-operation-parameters>
```

Response Body

When the request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-operation-zone

This module enables you to expunge or rename a zone object.

This expunge or rename action stages the modifications in the defined-configuration. To persist the staged modifications, save the changes using one of the following brocade-zone/effective-configuration container leafs:

- `cfg-action-v2` : Specify 'save' to save the pending changes.
- `cfg-name` : Specify the name of configuration to become enabled.

You must complete saving the changes within the configured fabric-lock-timeout duration specified in the brocade-fibrechannel-configuration/zone-configuration container.

Expunge deletes the object and removes the object from the member lists of all other objects. Expunge operations can be performed on cfgs, zones, aliases, WWNs, or 'D,I' members.

Renaming the object also replaces all occurrences of the renamed zone object in the entire defined zone database. Rename operations can be performed on cfgs, zones, aliases.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-operation-zone.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-operation-zone

  rpcs:
    +---x fibrechannel-zone
       +---w input
          +---w zone-operation-parameters* [] {fibrechannel:fibrechannel_switch_platform}?
             +---w action?            zone-operation-action-type
             +---w zone-object?       zone:zone-member-type
             +---w new-zone-object?   zone:zone-member-type
```

Examples

This section provides a basic example for using the brocade-operation-zone module.

Expunging a Zone Alias

This example uses a POST request to expunge a zone alias (ali30).

Expunge deletes the object and removes the object from the member lists of all other objects. Expunge operations can be performed on cfgs, zones, aliases, WWNs, or 'D,I' members.

Structure

```
POST base_URI/operations/fibrechannel-zone
```

URI

```
POST https://10.10.10.10/rest/operations/fibrechannel-zone
```

Request Body

```
<zone-operation-parameters>
   <action>expunge</action>
```

```
    <zone-object>ali30</zone-object>
</zone-operation-parameters>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status in the header.

Renaming a Zone Alias

This example uses a POST request to rename a zone (zone1).

Renaming the object also replaces all occurrences of the renamed zone object in the entire defined zone database. Rename operations can be performed on cfgs, zones, aliases.

Structure

```
POST base_URI/operations/fibrechannel-zone
```

URI

```
POST https://10.10.10.10/rest/operations/fibrechannel-zone
```

Request Body

```
<zone-operation-parameters>
   <action>rename</action>
   <zone-object>zone1</zone-object>
   <new-zone-object>zone1_renamed</new-zone-object>
</zone-operation-parameters>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status in the header.

FOS REST API Modules for Fibre Channel Features

This section details the FOS REST API support for Fibre Channel modules. This section also provides examples for using the FOS REST API Fibre Channel modules.

brocade-access-gateway

This module provides a detailed view of configuration and runtime information for the Access Gateway (AG).

For a list of changes to this module by release, refer to the individual module YANG file.

It assumes a knowledge of the Access Gateway as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Access Gateway User Guide*.

NOTE
The brocade-access-gateway module is supported in Fabric OS v9.0.x and later.

Module Tree

This is the tree view of the module from the `brocade-access-gateway.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-access-gateway
  +--rw brocade-access-gateway
     +--rw port-group* [port-group-id] {fibrechannel:access_gateway_platform}?
     |  +--rw port-group-id         uint8
     |  +--rw port-group-name?      string
     |  +--rw port-group-n-ports
     |  |  +--rw n-port*   fibrechannel:slot-port-name-type
     |  +--rw port-group-f-ports
     |  |  +--rw f-port*   fibrechannel:slot-port-name-type
     |  +--rw port-group-mode
     |     x--rw load-balancing-mode-enabled?                      uint8
     |     +--rw load-balancing-mode-enabled-v2?                   boolean
     |     x--rw multiple-fabric-name-monitoring-mode-enabled?     uint8
     |     +--rw multiple-fabric-name-monitoring-mode-enabled-v2?  boolean
     +--rw n-port-map* [n-port] {fibrechannel:access_gateway_platform}?
     |  +--rw n-port                    fibrechannel:slot-port-name-type
     |  x--rw failover-enabled?         uint8
     |  +--rw failover-enabled-v2?      boolean
     |  x--rw failback-enabled?         uint8
     |  +--rw failback-enabled-v2?      boolean
     |  x--ro online-status?            uint8
     |  +--ro operational-status?       access-gateway-type:operational-status-type
     |  x--ro reliable-status?          int8
     |  +--ro reliable-status-string?   access-gateway-type:reliable-type
     |  +--ro n-port-info
     |  |  +--ro attached-fabric-name?                fibrechannel:wwn-type
     |  |  +--ro attached-port-wwn?                   fibrechannel:wwn-type
     |  |  +--ro n-port-fcid?                         fibrechannel:fcid-hex-string-type
     |  |  +--ro attached-switch-user-friendly-name?  string
     |  |  +--ro attached-switch-f-port?              fibrechannel:slot-port-name-type
     |  |  +--ro attached-switch-ip-address?          inet:ip-address
     |  +--rw configured-f-port-list
     |  |  +--rw f-port*   fibrechannel:slot-port-name-type
     |  +--rw static-f-port-list
     |  |  +--rw f-port*   fibrechannel:slot-port-name-type
     |  +--rw preferred-f-ports
     |     +--rw preferred-f-port*   fibrechannel:slot-port-name-type
     +--ro f-port-list* [f-port] {fibrechannel:access_gateway_platform}?
```

```
|  +--ro f-port               fibrechannel:slot-port-name-type
|  x--ro online-status?       uint8
|  +--ro operational-status?  access-gateway-type:operational-status-type
|  +--ro f-port-info
|     +--ro n-port?             fibrechannel:slot-port-name-type
|     x--ro login-exceeded?     uint8
|     +--ro login-exceeded-v2?  boolean
+--rw policy {fibrechannel:access_gateway_platform}?
|  x--rw port-group-policy-enabled?     uint8
|  +--rw port-group-policy-enabled-v2?  boolean
|  x--rw auto-policy-enabled?           uint8
|  +--rw auto-policy-enabled-v2?        boolean
+--rw n-port-settings {fibrechannel:access_gateway_platform}?
|  +--rw reliability-counter?   yang:zero-based-counter64
+--ro device-list* [wwn] {fibrechannel:access_gateway_platform}?
   +--ro wwn       fibrechannel:wwn-type
   +--ro fcid?     fibrechannel:fcid-hex-string-type
   +--ro f-port?   fibrechannel:slot-port-name-type
   +--ro n-port?   fibrechannel:slot-port-name-type
```

Examples

This section provides basic examples for using the brocade-access-gateway module.

Viewing the Access Gateway Mode

You use a GET request to determine the Access Gateway mode of the switch. There are three possible modes:

- **0:** Access Gateway mode is not supported by this switch.
- **1:** Access Gateway mode is supported and currently disabled on this switch.
- **3:** Access Gateway mode is supported and currently enabled on this switch.

You must use a PATCH request to enable or disable Access Gateway mode on a switch.

The following example uses the GET request to determine the Access Gateway mode of the switch.

Structure

```
GET <base_URI>/running/brocade-switch/fibrechannel-switch/name/switch-worldwide-name/ag-mode
```

URI

```
GET https://10.10.10.10/rest/running/brocade-switch/fibrechannel-switch/name/10:10:10:eb:1a:b7:77:bc/ag-mode
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-switch>
        <name>10:10:10:eb:1a:b7:77:bc</name>
        <ag-mode>3</ag-mode>
```

```
    </fibrechannel-switch>
</Response>
```

Enabling Access Gateway Mode on a Switch

NOTE
You must disable the switch before you change the Access Gateway mode on a switch.

You must use a PATCH request to enable or disable the Access Gateway mode on a switch.

You use a GET request to determine the Access Gateway mode of the switch. There are three possible modes:

- **0:** Access Gateway mode is not supported by this switch.
- **1:** Access Gateway mode is supported and currently disabled on this switch.
- **3:** Access Gateway mode is supported and currently enabled on this switch.

The following example uses the PATCH request to enable Access Gateway mode on a switch.

Structure

```
PATCH <base_URI>/running/brocade-switch/fibrechannel-switch/name/fibre-channel-interface-name/ag-mode/mode
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-switch/fibrechannel-switch/name/10:10:10:eb:1a:b7:77:bc/ag-
mode/3
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Creating a New Port Group

The following example uses the POST request to create a new port group "pg1" with two N_Ports.

Structure

```
POST <base_URI>/running/brocade-access-gateway/port-group/port-group-id/port-group-id
```

URI

```
POST https://10.10.10.10/rest/running/brocade-access-gateway/port-group/port-group-id/1
```

Request Body

```
<port-group-n-ports>
    <n-port>0/40</n-port>
    <n-port>0/41</n-port>
</port-group-n-ports>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

Adding an N_Port to a Port Group

The following example uses the POST request to add a new N_Port "0/42" to an existing port group "pg1".

Structure

```
POST <base_URI>/running/brocade-access-gateway/port-group/port-group-id/port-group-id
```

URI

```
POST https://10.10.10.10/rest/running/brocade-access-gateway/port-group/port-group-id/1
```

Request Body

```
<port-group-n-ports>
    <n-port>0/42</n-port>
</port-group-n-ports>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

Adding an F_Port to a Port Group

The following example uses the POST request to add two new F_Ports to an existing port group "pg1".

Note that you can only add F-ports to a port-group when load-balancing mode is enabled (load-balancing-mode-enabled = 1).

Structure

```
POST <base_URI>/running/brocade-access-gateway/port-group/port-group-id/port-group-id
```

URI

```
POST https://10.10.10.10/rest/running/brocade-access-gateway/port-group/port-group-id/1
```

Request Body

```
<port-group-f-ports>
    <f-port>0/5</f-port>
    <f-port>0/6</f-port>
</port-group-f-ports>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

Enabling Load-Balancing Mode on a Port Group

The following example uses the PATCH request to enable load-balancing mode on port group "pg1".

Structure

```
PATCH <base_URI>/running/brocade-access-gateway/port-group/port-group-id/port-group-id/port-group-mode/port-group-mode
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-access-gateway/port-group/port-group-id/1/port-group-mode/
```

Request Body

```
<port-group-mode>
    <load-balancing-mode-enabled>1</load-balancing-mode-enabled>
</port-group-mode>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Enabling Multiple Fabric Name Monitor Mode on a Port Group

The following example uses the PATCH request to enable multiple fabric name monitor mode on port group "pg1".

Structure

```
PATCH <base_URI>/running/brocade-access-gateway/port-group/port-group-id/port-group-id/port-group-mode/
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-access-gateway/port-group/port-group-id/1/port-group-mode/
```

Request Body

```
<port-group-mode>
    <multiple-fabric-name-monitoring-mode-enabled>1</multiple-fabric-name-monitoring-mode-enabled>
</port-group-mode>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Deleting Ports From a Port Group

The following example uses the DELETE request to delete F_Port 0/5 and N_Port 0/42 from port group pg1.

Note that you can only remove F-ports from a port-group when load-balancing mode is enabled (load-balancing-mode-enabled = 1).

Structure

```
DELETE <base_URI>/running/brocade-access-gateway/port-group/port-group-id/port-group-id
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-access-gateway/port-group/port-group-id/1
```

Request Body

```
<port-group-f-ports>
   <f-port>0/5</f-port>
</port-group-f-ports>

<port-group-n-ports>
    <n-port>0/42</n-port>
</port-group-n-ports>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Deleting a Port Group

The following example uses the DELETE request to delete port group pg1.

Structure

```
DELETE <base_URI>/running/brocade-access-gateway/port-group/port-group-id/port-group-id
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-access-gateway/port-group/port-group-id/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Mapping an F_Port to an N_Port

The following example uses the PATCH request to map F_Ports 0/5 and 0/6 to N_Port 0/42.

NOTE
The port group policy must be enabled (port-group-policy-enabled = 1) to map F_Ports to an N_Port.

Structure

```
PATCH <base_URI>/running/brocade-access-gateway/n-port-map
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-access-gateway/n-port-map
```

Request Body

```
<n-port-map>
   <n-port>0/42</n-port>
   <configured-f-port-list>
      <f-port>0/5</f-port>
      <f-port>0/6</f-port>
   </configured-f-port-list>
</n-port-map>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Deleting an F_Port From an N_Port Mapping

The following example uses the DELETE request to remove the F_Port 0/6 to N_Port 0/42 mapping.

Structure

```
DELETE <base_URI>/running/brocade-access-gateway/n-port-map
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-access-gateway/n-port-map
```

Request Body

```
<n-port-map>
   <n-port>0/42</n-port>
   <configured-f-port-list>
      <f-port>0/6</f-port>
   </configured-f-port-list>
</n-port-map>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Viewing End Device Information

The following example uses the GET request to retrieve the N_Port login information and the attached switch details.

Structure

```
GET <base_URI>/running/brocade-access-gateway/n-port-map
```

URI

```
GET https://10.10.10.10/rest/running/brocade-access-gateway/n-port-map
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<Response>
    <n-port-map>
        <n-port>0/17</n-port>
        <failover-enabled>1</failover-enabled>
        <failback-enabled>1</failback-enabled>
        <online-status>1</online-status>
        <reliable-status>1</reliable-status>
        <n-port-info>
            <attached-fabric-name>10:10:10:05:1e:ab:47:00</attached-fabric-name>
            <attached-port-wwn>2f:03:c4:f5:7c:c4:da:25</attached-port-wwn>
            <n-port-fcid>0xec1300</n-port-fcid>
            <attached-switch-user-friendly-name>G610_066_236</attached-switch-user-friendly-name>
            <attached-switch-f-port>0/19</attached-switch-f-port>
            <attached-switch-ip-address>10.10.10.10</attached-switch-ip-address>
        </n-port-info>
        <configured-f-port-list/>
        <static-f-port-list/>
        <preferred-f-port-list/>
    </n-port-map>
...
    <n-port-map>
        <n-port>0/21</n-port>
        <failover-enabled>1</failover-enabled>
        <failback-enabled>1</failback-enabled>
        <online-status>1</online-status>
        <reliable-status>1</reliable-status>
        <n-port-info>
            <attached-fabric-name>10:10:10:05:1e:ab:47:00</attached-fabric-name>
            <attached-port-wwn>20:19:c4:f5:7c:00:c5:30</attached-port-wwn>
            <n-port-fcid>0xe81900</n-port-fcid>
            <attached-switch-user-friendly-name>G630_066_232</attached-switch-user-friendly-name>
            <attached-switch-f-port>0/25</attached-switch-f-port>
            <attached-switch-ip-address>10.10.10.10</attached-switch-ip-address>
        </n-port-info>
        <configured-f-port-list>
            <f-port>0/9</f-port>
            <f-port>0/10</f-port>
            <f-port>0/11</f-port>
            <f-port>0/12</f-port>
            <f-port>0/13</f-port>
            <f-port>0/14</f-port>
            <f-port>0/15</f-port>
        </configured-f-port-list>
        <static-f-port-list/>
        <preferred-f-port-list/>
    </n-port-map>
</Response>
```

Configuring an N_Port

The following example uses the PATCH request to configure port 0/45 to operate as an N_Port.

Structure

```
PATCH <base_URI>/running/brocade-interface/fibrechannel/name/fibre-channel-interface-name/n-port-enabled/1
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/0%2f45/n-port-enabled/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Configuring the Reliability Counter for N_Ports

The following example uses the PATCH request to configure the reliability counter to 30.

NOTE
The port group policy must be enabled (port-group-policy-enabled = 1) to configure the reliability counter.

Structure

```
PATCH <base_URI>/running/brocade-access-gateway/n-port-settings/reliability-counter/reliability-counter-value
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-access-gateway/n-port-settings/reliability-counter/30
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Enabling Failover for an N_Port

The following example uses the PATCH request to enable failover for N_Port 0/40.

Structure

```
PATCH <base_URI>/running/brocade-access-gateway/n-port-map
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-access-gateway/n-port-map
```

Request Body

```
<n-port-map>
  <n-port>0/40</n-port>
```

```
  <failover-enabled>1</failover-enabled>
</n-port-map>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Enabling Failback for an N_Port

The following example uses the PATCH request to enable failback for N_Port 0/42.

Structure

```
PATCH <base_URI>/running/brocade-access-gateway/n-port-map
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-access-gateway/n-port-map
```

Request Body

```
<n-port-map>
  <n-port>0/42</n-port>
  <failback-enabled>1</failback-enabled>
</n-port-map>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Viewing the List of Devices Logged On to the Access Gateway Switch

The following example uses a GET request to view a list of devices logged on to the Access Gateway switch and the device's mapping with the F_Port, N_Port, and FCID.

Structure

```
GET <base_URI>/running/brocade-access-gateway/device-list
```

URI

```
GET https://10.10.10.10/rest/running/brocade-access-gateway/device-list
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <device-list>
        <wwn>10:10:10:eb:1a:b7:77:bc</wwn>
```

```
        <fcid>0x021002</fcid>
        <f-port>0/3</f-port>
        <n-port>0/16</n-port>
    </device-list>
</Response>
```

brocade-application-server

This module enables you to monitor application-specific services.

For a list of changes to this module by release, refer to the individual module YANG file.

The brocade-application-server module supports the following query parameter combinations:

- n-port-id
- n-port-id + application-id
- entity-id-ascii
- n-port-id + entity-id-ascii

Module Tree

This is the tree view of the module from the brocade-application-server.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-application-server
  +--ro brocade-application-server
     +--ro application-server-device* [] {fibrechannel:fibrechannel_switch_platform}?
        +--ro n-port-id?         fibrechannel:fcid-hex-string-type
        +--ro application-id?    uint32
        +--ro entity-id-hex?     fibrechannel:brocade-hex-string-type
        +--ro entity-id-ascii?   string
        +--ro entity-name?       string
        +--ro host-identifier?   string
        +--ro symbolic-data?     string
```

Examples

This section provides basic examples for using the brocade-application-server module.

Viewing the Application Server Information for an Entity ID

This example uses a GET request to view the application server information for a specified entity ID.

Structure

```
GET <base_URI>/running/brocade-application-server/application-server-device/?entity-id-ascii={eid}
```

URI

```
GET https://10.10.10.10/rest/running/brocade-application-server/application-server-device/?entity-id-
ascii=11%3A22%3A33%3A44%3A55%3A66%3A77%3A88 - 99%3AAA%3ABB%3ACC%3ADD%3AEE%3AFF%3A00
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
{
    "Response": {
        "application-server-device": [
            {
```

```
                "n-port-id": "0x010e00",
                "application-id": 521,
                "entity-id-hex":
 "0x313120323220333320343420353520363620373720383820 2d20393920414120424220434320444420454520464620 3030",
                "entity-id-ascii": "11 22 33 44 55 66 77 88 - 99 AA BB CC DD EE FF 00"
                "entity-name": "My_VM_Instance"
                "host-identifier": "My_Department_ESXi_Host"
                "symbolic-data": "VMware ESXi 7.0 GA [build 15843807]"

            }
        ]
    }
}
```


Viewing the Application Server Information for an N_Port ID

This example uses a GET request to view the application server information for a specified N_Port ID.

Structure

```
GET <base_URI>/running/brocade-application-server/application-server-device/?n-port-id={n_port_id}
```

URI

```
GET https://10.10.10.10/rest/running/brocade-application-server/application-server-device/?n-port-id=0x010e00
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.


```
{
    "Response": {
        "application-server-device": [
            {
                "n-port-id": "0x010e00",
                "application-id": 521,
                "entity-id-hex":
 "0x313120323220333320343420353520363620373720383820 2d20393920414120424220434320444420454520464620 3030",
                "entity-id-ascii": "11 22 33 44 55 66 77 88 - 99 AA BB CC DD EE FF 00"
                "entity-name": "My_VM_Instance"
                "host-identifier": "My_Department_ESXi_Host"
                "symbolic-data": "VMware ESXi 7.0 GA [build 15843807]"

            }
        ]
    }
}
```

brocade-chassis

This module provides a detailed view of configuration and runtime information for the Fabric OS switch or director.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-chassis.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-chassis
  +--rw brocade-chassis
     +--rw chassis
     |  +--rw chassis-user-friendly-name?      fibrechannel:chassis-user-friendly-name-type
     |  +--ro chassis-wwn?                     fibrechannel:wwn-type
     |  +--ro license-id?                      fibrechannel:wwn-type
     |  +--ro manufacturer?                    fru:manufacturer-type
     |  +--ro registered-organization-name?    string
     |  +--ro organization-registration-date?  string
     |  x--ro part-number?                     fru:part-number-type
     |  +--ro serial-number?                   fru:serial-number-type
     |  +--ro entitlement-serial-number?       fru:serial-number-type
     |  +--ro max-blades-supported?            uint16
     |  x--ro vendor-serial-number?            string
     |  x--ro vendor-part-number?              string
     |  x--ro vendor-revision-number?          string
     |  +--ro product-name?                    fibrechannel:product-name-type
     |  +--rw vf-enabled?                      boolean {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro vf-supported?                    boolean {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw fcr-enabled?                     boolean
     |  +--ro fcr-supported?                   boolean {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro date?                            string
     |  +--rw chassis-enabled?                 boolean
     |  +--rw message-of-the-day?              string
     |  +--rw shell-timeout?                   uint32
     |  +--rw session-timeout?                 uint32
     |  +--rw usb-device-enabled?              boolean
     |  +--ro usb-available-space?             uint32
     |  +--rw tcp-timeout-level?               uint8
     |  +--ro backplane-revision?              fibrechannel:brocade-hex-string-type
  {fibrechannel:fibrechannel_chassis_platform}?
     |  +--ro system-uptime?                   yang:timeticks
     |  +--ro user-sessions?                   uint32
     |  +--ro one-minute-load-average?         decimal64
     |  +--ro five-minutes-load-average?       decimal64
     |  +--ro fifteen-minutes-load-average?    decimal64
     |  +--ro total-memory?                    yang:gauge64
     |  +--ro used-memory?                     yang:gauge64
     |  +--ro free-memory?                     yang:gauge64
     |  +--ro shared-memory?                   yang:gauge64
     |  +--ro buffer-cache-memory?             yang:gauge64
     |  +--ro available-memory?                yang:gauge64
     +--rw version
```

```
|  +--rw kernel?                          string
|  +--rw fabric-os?                       string
|  +--rw made-on?                         yang:date-and-time
|  +--rw flash?                           yang:date-and-time
|  +--rw boot-prom?                       string
+--ro ha-status
|  +--ro active-cp?                       string
|  +--ro standby-cp?                      string
|  +--ro active-slot?                     uint16
|  +--ro standby-slot?                    uint16
|  +--ro recovery-type?                   string
|  +--ro standby-health?                  string
|  +--ro ha-enabled?                      boolean
|  +--ro heartbeat-up?                    boolean
|  +--ro ha-synchronized?                 boolean
+--rw management-interface-configuration
|  +--rw rest-enabled?                    boolean
|  +--rw https-protocol-enabled?          boolean
|  +--ro effective-protocol?              enumeration
|  +--rw max-rest-sessions?               uint32
|  +--rw https-keep-alive-enabled?        boolean
|  +--ro https-keep-alive-timeout?        uint32
+--rw management-ethernet-interface* [cp-name interface-name]
|  +--rw cp-name                          fibrechannel:cp-name-type
|  +--rw interface-name                   enumeration
|  +--rw auto-negotiate?                  boolean
|  +--rw speed?                           enumeration
|  +--ro duplex?                          enumeration
|  +--ro connection-established-status?   boolean
|  +--ro ethernet-mac-address?            yang:mac-address
|  +--ro active-interface?                union {fibrechannel:fibrechannel_chassis_platform}?
|  +--ro standby-interfaces {fibrechannel:fibrechannel_chassis_platform}?
|  |  +--ro interface*                    fibrechannel:physical-interface-type
|  +--ro inet-address?                    inet:ipv4-address
|  +--ro broadcast-address?               inet:ipv4-address
|  +--ro subnet-mask?                     inet:ipv4-address
|  +--ro global-inet6-addresses
|  |  +--ro global-inet6-address*         union
|  +--ro link-inet6-address?              union
|  +--ro ethernet-status-flags
|  |  +--ro flag*   enumeration
|  +--ro mtu?                             uint32
|  +--ro metric?                          yang:gauge32
|  +--ro rx-packets?                      yang:zero-based-counter64
|  +--ro rx-errors?                       yang:zero-based-counter64
|  +--ro rx-dropped?                      yang:zero-based-counter64
|  +--ro rx-overruns?                     yang:zero-based-counter64
|  +--ro rx-frame?                        yang:zero-based-counter64
|  +--ro rx-bytes?                        yang:zero-based-counter64
|  +--ro tx-packets?                      yang:zero-based-counter64
|  +--ro tx-errors?                       yang:zero-based-counter64
|  +--ro tx-dropped?                      yang:zero-based-counter64
|  +--ro tx-overruns?                     yang:zero-based-counter64
```

```
|  +--ro tx-carrier?                     yang:zero-based-counter64
|  +--ro tx-collisions?                  yang:zero-based-counter64
|  +--ro tx-queue-length?                uint32
|  +--ro tx-bytes?                       yang:zero-based-counter64
|  +--ro start-memory-address?           fibrechannel:brocade-hex-string-type
|  +--ro end-memory-address?             fibrechannel:brocade-hex-string-type
|  +--rw lldp-enabled?                   boolean
+--rw management-port-track-configuration
|  +--rw tracking-enabled?               boolean
|  +--rw scan-interval?                  uint16
+--ro management-port-connection-statistics* []
|  +--ro cp-name?                        fibrechannel:cp-name-type
|  +--ro ip-family?                      inet:ip-version
|  +--ro operational-status?             enumeration
|  +--ro local-ip-address?               inet:ip-address
|  +--ro local-port?                     inet:port-number
|  +--ro remote-ip-address?              inet:ip-address
|  +--ro remote-port?                    inet:port-number
|  +--ro start-time?                     yang:date-and-time
|  +--ro state-update-time?              union
|  +--ro protocol?                       string
|  +--ro connection-state?               brocade-interface-types:connection-state-type
|  +--ro process-name?                   string
|  +--ro process-id?                     uint16
+--rw credit-recovery {fibrechannel:fibrechannel_chassis_platform}?
   +--rw mode?                         enumeration
   +--rw link-reset-threshold?         uint16
   +--rw fault-option?                 enumeration
   +--rw backend-credit-loss-enabled?  boolean
   +--rw backend-loss-of-sync-enabled? boolean
```

Examples

This section provides basic examples for using the brocade-chassis module.

Viewing the Chassis Name

The following example uses the GET request to view the administrator-assigned name of the chassis.

Structure

```
GET <base_URI>/running/brocade-chassis/chassis/chassis-user-friendly-name
```

URI

```
GET https://10.10.10.10/rest/running/brocade-chassis/chassis/chassis-user-friendly-name
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
```

```
    <chassis>
        <chassis-user-friendly-name>BrocadeX6-4</chassis-user-friendly-name>
    </chassis>
</Response>
```

Viewing the Chassis World Wide Name

The following example uses the GET request to view the WWN the chassis.

Structure

```
GET <base_URI>/running/brocade-chassis/chassis/chassis-wwn
```

URI

```
GET https://10.10.10.10/rest/running/brocade-chassis/chassis/chassis-wwn
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <chassis>
        <chassis-wwn>10:00:c1:f1:1c:1a:ac:eb</chassis-wwn>
    </chassis>
</Response>
```

Enabling Virtual Fabrics on the Chassis

The following example uses the PATCH request to enable Virtual Fabrics on the chassis.

To enable a specific Virtual Fabric, refer to the brocade-fibrechannel-logical-switch.

NOTE
Enabling Virtual Fabrics on a chassis is a disruptive operation and the device automatically reboots after running this request.

Structure

```
PATCH <base_URI>/running/brocade-chassis/chassis/vf-enabled/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-chassis/chassis/vf-enabled/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Viewing the High Availability Status

The following example uses the GET request to view the High Availability (HA) status .

Structure

```
GET <base_URI>/running/brocade-chassis/ha-status
```

URI

```
GET https://10.10.10.10/rest/running/brocade-chassis/ha-status
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <ha-status>
        <active-cp>cp0</active-cp>
        <standby-cp>cp1</standby-cp>
        <active-slot>1</active-slot>
        <standby-slot>2</standby-slot>
        <recovery-type>warm</recovery-type>
        <standby-health>healthy</standby-health>
        <ha-enabled>true</ha-enabled>
        <heartbeat-up>true</heartbeat-up>
        <ha-synchronized>true</ha-synchronized>
    </ha-status>
</Response>
```

Determining if HA Is Enabled

The following example uses the GET request to determine the HA status of the switch.

Structure

```
GET <base_URI>/running/brocade-chassis/ha-status/ha-enabled
```

URI

```
GET https://10.10.10.10/rest/running/brocade-chassis/ha-status/ha-enabled
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <ha-status>
        <ha-enabled>true</ha-enabled>
    </ha-status>
</Response>
```

Determining the HA Synchronized Status

The following example uses the GET request to determine whether HA is synchronized between the active and standby CPs.

Structure

```
GET <base_URI>/running/brocade-chassis/ha-status/ha-synchronized
```

URI

```
GET https://10.10.10.10/rest/running/brocade-chassis/ha-status/ha-synchronized
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <ha-status>
        <ha-synchronized>false</ha-synchronized>
    </ha-status>
</Response>
```

brocade-extension

This module enables you to view information about Brocade extension objects as well as to configure traffic control lists.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-extension.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-extension
  +--rw brocade-extension
     +--ro dp-hcl-status* [slot dp-id] {fibrechannel:fibrechannel_extension_platform}?
     |  +--ro slot                        uint16
     |  +--ro dp-id                       extension:dp-id-type
     |  +--ro state?                      extension:dp-state-type
     |  +--ro status?                     extension:dp-actstate-type
     |  +--ro fc-hcl-stage?               extension:dp-hcl-stage-type
     |  +--ro ip-hcl-stage?               extension:dp-hcl-stage-type
     |  +--ro svi-swapped?                boolean
     |  +--ro dp-communication-status?    extension:dp-dp-communication-type
     |  +--ro firmware-version?           string
     +--rw traffic-control-list* [traffic-control-list-name] {fibrechannel:fibrechannel_extension_platform}?
     |  +--rw traffic-control-list-name            extension:extension-name-type
     |  +--rw priority?                            uint32
     |  +--rw admin-state-enabled?                 boolean
     |  +--ro cp-dp-synchronized?                  boolean
     |  +--rw action?                              extension:tcl-action-type
     |  +--rw target-ve-port?                      extension:ve-interface-type
     |  +--rw target-qos?                          extension:ip-qos-type
     |  +--rw target-slot?                         uint16
     |  +--rw target-dp-id?                        extension:dp-id-type
     |  +--rw source-address?                      union
     |  +--rw source-address-prefix-length?        brocade-interface-types:ip-prefix-length
     |  +--rw destination-address?                 union
     |  +--rw destination-address-prefix-length?   brocade-interface-types:ip-prefix-length
     |  +--rw port?                                string
     |  +--rw application?                         extension:extension-name-type
     |  +--rw dscp?                                string
     |  +--rw vlan?                                string
     |  +--rw l2cos?                               string
     |  +--rw l4-protocol?                         extension:l4-protocol-type
     |  +--rw reset-propagation-enabled?           boolean
     |  +--rw segment-preservation-enabled?        boolean
     |  +--rw non-terminated-enabled?              boolean
     |  +--ro hit-count?                           yang:zero-based-counter64
     +--ro lan-flow-statistics* [flow-index slot dp-id] {fibrechannel:fibrechannel_extension_platform}?
     |  +--ro flow-index                          uint32
     |  +--ro slot                                uint16
     |  +--ro dp-id                               extension:dp-id-type
     |  +--ro protocol?                           extension:l4-protocol-type
     |  +--ro source-port?                        inet:port-number
     |  +--ro destination-port?                   inet:port-number
```

```
|  +--ro source-ip-address?                inet:ip-address
|  +--ro destination-ip-address?           inet:ip-address
|  +--ro traffic-control-list-name?        extension:extension-name-type
|  +--ro lan-interface?                    extension:extension-lan-interface-type
|  +--ro mapped-tunnel
|  |  +--ro ve-port?   extension:ve-interface-type
|  |  +--ro qos?       extension:ip-qos-type
|  +--ro vlan-id?                          extension:vlan-id-type
|  +--ro vlan-priority?                    uint32
|  +--ro dscp?                             inet:dscp
|  +--ro hcl-flow?                         boolean
|  +--ro active-flow?                      boolean
|  +--ro start-time?                       yang:date-and-time
|  +--ro end-time?                         yang:date-and-time
|  x--ro rtt?                              yang:gauge32
|  +--ro tcp-rtt?                          yang:gauge64
|  +--ro in-tcp-bytes?                     yang:zero-based-counter64
|  +--ro in-tcp-packets?                   yang:zero-based-counter64
|  +--ro out-tcp-bytes?                    yang:zero-based-counter64
|  +--ro out-tcp-packets?                  yang:zero-based-counter64
|  +--ro tcp-out-of-order-packets?         yang:zero-based-counter64
|  +--ro tcp-retransmits?                  yang:zero-based-counter64
|  +--ro fast-retransmits?                 yang:zero-based-counter64
|  +--ro slow-retransmits?                 yang:zero-based-counter64
|  +--ro duplicate-acknowledgement?        yang:zero-based-counter64
|  +--ro zero-window-count?                yang:zero-based-counter64
|  +--ro zero-window-maximum-duration?     yang:zero-based-counter64
|  +--ro out-bytes-average?                yang:gauge64
|  +--ro in-bytes-average?                 yang:gauge64
|  +--ro in-drops-lan-session-manager?     yang:zero-based-counter64
|  +--ro out-drops-lan-session-manager?    yang:zero-based-counter64
|  +--ro crc-errors?                       yang:zero-based-counter64
|  +--ro in-bytes-lan-session-manager?     yang:zero-based-counter64
|  +--ro in-packets-lan-session-manager?   yang:zero-based-counter64
|  +--ro out-bytes-lan-session-manager?    yang:zero-based-counter64
|  +--ro out-packets-lan-session-manager?  yang:zero-based-counter64
|  +--ro compression-enabled?              boolean
|  +--ro in-bytes-lan-compression?         yang:zero-based-counter64
|  +--ro in-bytes-wan-compression?         yang:zero-based-counter64
|  +--ro out-bytes-lan-compression?        yang:zero-based-counter64
|  +--ro out-bytes-wan-compression?        yang:zero-based-counter64
|  +--ro local-host-mss?                   yang:zero-based-counter32
|  +--ro remote-host-mss?                  yang:zero-based-counter32
+--ro global-lan-statistics* [slot dp-id] {fibrechannel:fibrechannel_extension_platform}?
   +--ro slot                                                uint16
   +--ro dp-id                                               extension:dp-id-type
   +--ro active-tcp-connections?                             yang:gauge32
   +--ro active-tcp-connections-on-local-backup?             yang:gauge32
   +--ro active-tcp-connections-on-remote-backup?            yang:gauge32
   +--ro established-tcp-connections?                        yang:gauge32
   +--ro closed-tcp-connections?                             yang:gauge32
   +--ro out-tcp-bytes?                                      yang:zero-based-counter64
   +--ro in-tcp-bytes?                                       yang:zero-based-counter64
```

```
+--ro tcp-tcl-deny-connections?                                yang:zero-based-counter64
+--ro tcp-tcl-lookup-fail?                                     yang:zero-based-counter64
+--ro syn-received?                                            yang:zero-based-counter64
+--ro syn-fail?                                                yang:zero-based-counter64
+--ro drop-bytes?                                              yang:zero-based-counter64
+--ro drop-packets?                                            yang:zero-based-counter64
+--ro stale-reset-from-host?                                   yang:zero-based-counter64
+--ro maximum-tcp-connection-exceeded-as-client?              yang:zero-based-counter64
+--ro maximum-tcp-connection-exceeded-as-server?              yang:zero-based-counter64
+--ro maximum-tcp-connection-per-second-exceeded-as-client?   yang:zero-based-counter64
+--ro maximum-tcp-connection-per-second-exceeded-as-server?   yang:zero-based-counter64
o--ro maximum-udp-connection-exceeded?                        yang:zero-based-counter64
+--ro maximum-udp-connection-exceeded-on-egress?              yang:zero-based-counter64
+--ro udp-packets-sent-as-is?                                 yang:zero-based-counter64
+--ro out-pdu-preserve-on?                                    yang:zero-based-counter64
+--ro in-pdu-preserve-on?                                     yang:zero-based-counter64
+--ro total-ipv6-packets?                                     yang:zero-based-counter64
+--ro flow-control-on?                                        yang:zero-based-counter64
+--ro flow-control-off?                                       yang:zero-based-counter64
+--ro active-udp-connections?                                 yang:zero-based-counter64
+--ro established-udp-connections?                            yang:zero-based-counter64
+--ro closed-udp-connections?                                 yang:zero-based-counter64
+--ro udp-route-lookup-fail?                                  yang:zero-based-counter64
+--ro udp-pdu-drops-due-to-pko-flow-control?                  yang:zero-based-counter64
+--ro out-udp-pdus?                                           yang:zero-based-counter64
+--ro out-udp-pdu-drops?                                      yang:zero-based-counter64
+--ro in-udp-pdus?                                            yang:zero-based-counter64
+--ro in-udp-tcl-lookup-fail-pdus?                            yang:zero-based-counter64
+--ro in-udp-tcl-deny-pdus?                                   yang:zero-based-counter64
+--ro in-total-udp-pdu-drops?                                 yang:zero-based-counter64
+--ro in-udp-pdu-drops-due-to-stream-flow-control?            yang:zero-based-counter64
+--ro out-icmp-pdus?                                          yang:zero-based-counter64
+--ro out-icmp-pdu-drops?                                     yang:zero-based-counter64
+--ro in-icmp-pdus?                                           yang:zero-based-counter64
+--ro in-icmp-tcl-lookup-fail-pdus?                           yang:zero-based-counter64
+--ro in-icmp-tcl-deny-pdus?                                  yang:zero-based-counter64
+--ro in-total-icmp-pdu-drops?                                yang:zero-based-counter64
+--ro in-icmp-pdu-drops-due-to-stream-flow-control?           yang:zero-based-counter64
+--ro out-as-is-ip-pdus?                                      yang:zero-based-counter64
+--ro out-as-is-ip-pdu-drops?                                 yang:zero-based-counter64
+--ro in-as-is-ip-pdus?                                       yang:zero-based-counter64
+--ro in-as-is-ip-tcl-lookup-fail-pdus?                       yang:zero-based-counter64
+--ro in-as-is-ip-tcl-deny-pdus?                              yang:zero-based-counter64
+--ro in-total-as-is-ip-pdu-drops?                            yang:zero-based-counter64
+--ro in-as-is-ip-pdu-drops-due-to-stream-flow-control?       yang:zero-based-counter64
+--ro in-error-ip-checksum?                                   yang:zero-based-counter64
+--ro in-error-tcp-checksum?                                  yang:zero-based-counter64
+--ro in-error-mac?                                           yang:zero-based-counter64
+--ro in-error-crc?                                           yang:zero-based-counter64
+--ro in-error-parity?                                        yang:zero-based-counter64
+--ro in-error-length?                                        yang:zero-based-counter64
+--ro out-udp-packets-64-bytes?                               yang:zero-based-counter64
+--ro out-udp-packets-128-bytes?                              yang:zero-based-counter64
```

```
        +--ro out-udp-packets-256-bytes?                yang:zero-based-counter64
        +--ro out-udp-packets-512-bytes?                yang:zero-based-counter64
        +--ro out-udp-packets-1024-bytes?               yang:zero-based-counter64
        +--ro out-udp-packets-1500-bytes?               yang:zero-based-counter64
        +--ro out-udp-packets-3000-bytes?               yang:zero-based-counter64
        +--ro out-udp-packets-4500-bytes?               yang:zero-based-counter64
        +--ro out-udp-packets-6000-bytes?               yang:zero-based-counter64
        +--ro out-udp-packets-9000-bytes?               yang:zero-based-counter64
        +--ro in-udp-packets-64-bytes?                  yang:zero-based-counter64
        +--ro in-udp-packets-128-bytes?                 yang:zero-based-counter64
        +--ro in-udp-packets-256-bytes?                 yang:zero-based-counter64
        +--ro in-udp-packets-512-bytes?                 yang:zero-based-counter64
        +--ro in-udp-packets-1024-bytes?                yang:zero-based-counter64
        +--ro in-udp-packets-1500-bytes?                yang:zero-based-counter64
        +--ro in-udp-packets-3000-bytes?                yang:zero-based-counter64
        +--ro in-udp-packets-4500-bytes?                yang:zero-based-counter64
        +--ro in-udp-packets-6000-bytes?                yang:zero-based-counter64
        +--ro in-udp-packets-9000-bytes?                yang:zero-based-counter64
```

Examples

This section provides basic examples for using the brocade-extension module.

Viewing the Extension HCL Status

This example uses a GET request to view the HCL status for the Extension switch or blade.

Structure

```
GET <base_URI>/running/brocade-extension/dp-hcl-status/slot/slot_number/dp-id/dp_id
```

URI

```
GET https://10.10.10.10/rest/running/brocade-extension/dp-hcl-status/slot/8/dp-id/0
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <dp-hcl-status>
        <slot>8</slot>
        <dp-id>0</dp-id>
        <status>inactive</status>
        <state>online</state>
        <firmware-version>v9.0.0_bld83</firmware-version>
        <fc-hcl-stage>idle</fc-hcl-stage>
        <ip-hcl-stage>idle</ip-hcl-stage>
        <svi-swapped>false</svi-swapped>
        <dp-communication-status>up</dp-communication-status>
```

```
    </dp-hcl-status>
</Response>
```

Viewing the Traffic Control Lists

This example uses a GET request to view all traffic control lists configured on the Extension switch or blade.

Structure

```
GET <base_URI>/running/brocade-extension/traffic-control-list/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-extension/traffic-control-list/
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <traffic-control-list>
        <traffic-control-list-name>Test2</traffic-control-list-name>
        <priority>2</priority>
        <admin-state-enabled>true</admin-state-enabled>
        <cp-dp-synchronized>true</cp-dp-synchronized>
        <target-ve-port>8/17</target-ve-port>
        <target-qos>IP-priority-medium</target-qos>
        <vlan>any</vlan>
        <l2cos>any</l2cos>
        <dscp>any</dscp>
        <source-address>any</source-address>
        <destination-address>any</destination-address>
        <l4-protocol>any</l4-protocol>
        <port>any</port>
        <application>any</application>
        <action>allow</action>
        <reset-propagation-enabled>false</reset-propagation-enabled>
        <segment-preservation-enabled>false</segment-preservation-enabled>
        <non-terminated-enabled>false</non-terminated-enabled>
        <hit-count>13297</hit-count>
    </traffic-control-list>
    <traffic-control-list>
        <traffic-control-list-name>default</traffic-control-list-name>
        <priority>65535</priority>
        <admin-state-enabled>true</admin-state-enabled>
        <cp-dp-synchronized>true</cp-dp-synchronized>
        <vlan>any</vlan>
        <l2cos>any</l2cos>
        <dscp>any</dscp>
```

```
        <source-address>any</source-address>
        <destination-address>any</destination-address>
        <l4-protocol>any</l4-protocol>
        <port>any</port>
        <application>any</application>
        <action>deny</action>
        <reset-propagation-enabled>false</reset-propagation-enabled>
        <segment-preservation-enabled>false</segment-preservation-enabled>
        <non-terminated-enabled>false</non-terminated-enabled>
        <hit-count>23383</hit-count>
    </traffic-control-list>
</Response>
```

Configuring a Traffic Control List

This example uses a POST request to configure a traffic control list on the Extension switch or blade.

Structure

```
POST <base_URI>/running/brocade-extension/traffic-control-list
```

URI

```
POST https://10.10.10.10/rest/running/brocade-extension/traffic-control-list
```

Request Body

```
    <traffic-control-list>
        <traffic-control-list-name>Test2</traffic-control-list-name>
        <priority>2</priority>
        <admin-state-enabled>true</admin-state-enabled>
        <target-ve-port>8/17</target-ve-port>
        <target-qos>IP-priority-medium</target-qos>
        <vlan>any</vlan>
        <l2cos>any</l2cos>
        <dscp>any</dscp>
        <source-address>any</source-address>
        <destination-address>any</destination-address>
        <l4-protocol>any</l4-protocol>
        <port>any</port>
        <application>any</application>
        <action>allow</action>
        <reset-propagation-enabled>false</reset-propagation-enabled>
        <segment-preservation-enabled>false</segment-preservation-enabled>
        <non-terminated-enabled>false</non-terminated-enabled>
    </traffic-control-list>
```

Response Body

When the resource is successfully created, the response contains an empty message body and a “201 Created” status in the headers.

Enabling the Admin Status of a Traffic Control List

This example uses a PATCH request to enable the admin status of the Test3 traffic control list.

Structure

```
PATCH  <base_URI>/running/brocade-extension/traffic-control-list/<traffic-control-list-name>/rule0/admin-state-enabled/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-extension/traffic-control-list/traffic-control-list-name/rule0/admin-state-enabled/true
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Deleting a Traffic Control List

This example uses a DELETE request to delete the Test3 traffic control list..

Structure

```
DELETE <base_URI>/running/brocade-extension/traffic-control-list/<traffic_control_list_name>
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-extension/traffic-control-list/traffic-control-list-name/Test3
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the LAN Flow Statistics

This example uses a GET request to view the LAN flow statistics for the Extension switch or blade.

Structure

```
GET <base_URI>/running/running/brocade-extension/lan-flow-statistics/flow-index/flow_index/slot/slot_number/dp-id/dp_id
```

URI

```
GET https://10.10.10.10/rest/running/brocade-extension/lan-flow-statistics/flow-index/7/slot/0/dp-id/0
```

Request Body

There is no request body.

Response Body

```
<?xml version="1.0"?>
```

```
<Response>
    <lan-flow-statistics>
        <flow-index>7</flow-index>
        <slot>0</slot>
        <dp-id>0</dp-id>
        <protocol>TCP</protocol>
        <source-port>61840</source-port>
        <destination-port>60771</destination-port>
        <source-ip-address>174.10.6.5</source-ip-address>
        <destination-ip-address>165.105.21.5</destination-ip-address>
        <lan-interface>0/6</lan-interface>
        <mapped-tunnel>
            <ve-port>0/12</ve-port>
            <qos>IP-priority-high</qos>
        </mapped-tunnel>
        <vlan-id>0</vlan-id>
        <vlan-priority>0</vlan-priority>
        <dscp>0</dscp>
        <hcl-flow>false</hcl-flow>
        <active-flow>true</active-flow>
        <start-time>2020-01-29T17:56:09-08:00</start-time>
        <rtt>0</rtt>
        <in-tcp-bytes>189636844</in-tcp-bytes>
        <in-tcp-packets>126487</in-tcp-packets>
        <out-tcp-bytes>1176216</out-tcp-bytes>
        <out-tcp-packets>29405</out-tcp-packets>
        <tcp-out-of-order-packets>0</tcp-out-of-order-packets>
        <tcp-retransmits>0</tcp-retransmits>
        <fast-retransmits>0</fast-retransmits>
        <slow-retransmits>0</slow-retransmits>
        <duplicate-acknowledgement>0</duplicate-acknowledgement>
        <zero-window-count>5355</zero-window-count>
        <zero-window-maximum-duration>17</zero-window-maximum-duration>
        <in-bytes-average>0</in-bytes-average>
        <out-bytes-average>0</out-bytes-average>
        <in-drops-lan-session-manager>0</in-drops-lan-session-manager>
        <out-drops-lan-session-manager>0</out-drops-lan-session-manager>
        <in-bytes-lan-session-manager>184577352</in-bytes-lan-session-manager>
        <out-bytes-lan-session-manager>4</out-bytes-lan-session-manager>
        <in-packets-lan-session-manager>0</in-packets-lan-session-manager>
        <out-packets-lan-session-manager>1</out-packets-lan-session-manager>
        <crc-errors>0</crc-errors>
        <local-host-mss>1460</local-host-mss>
        <remote-host-mss>1460</remote-host-mss>
    </lan-flow-statistics>
```

Viewing the Global LAN Statistics

This example uses a GET request to view the global LAN statistics for the Extension switch or blade.

Structure

```
GET <base_URI>/running/brocade-extension/global-lan-statistics/slot/slot_number/dp-id/dp_id
```

URI

```
GET https://10.10.10.10/rest/running/brocade-extension/global-lan-statistics/slot/8/dp-id/0
```

Request Body

There is no request body.

Response Body

```
<Response>
    <global-lan-statistics>
        <slot>8</slot>
        <dp-id>0</dp-id>
        <active-tcp-connections>0</active-tcp-connections>
        <active-tcp-connections-on-local-backup>0</active-tcp-connections-on-local-backup>
        <active-tcp-connections-on-remote-backup>0</active-tcp-connections-on-remote-backup>
        <established-tcp-connections>0</established-tcp-connections>
        <closed-tcp-connections>0</closed-tcp-connections>
        <out-tcp-bytes>0</out-tcp-bytes>
        <in-tcp-bytes>0</in-tcp-bytes>
        <tcp-tcl-deny-connections>0</tcp-tcl-deny-connections>
        <tcp-tcl-lookup-fail>0</tcp-tcl-lookup-fail>
        <syn-received>0</syn-received>
        <syn-fail>0</syn-fail>
        <drop-bytes>0</drop-bytes>
        <drop-packets>0</drop-packets>
        <stale-reset-from-host>0</stale-reset-from-host>
        <maximum-tcp-connection-exceeded-as-client>0</maximum-tcp-connection-exceeded-as-client>
        <maximum-tcp-connection-exceeded-as-server>0</maximum-tcp-connection-exceeded-as-server>
        <maximum-tcp-connection-per-second-exceeded-as-client>0</maximum-tcp-connection-per-second-exceeded-
as-client>
        <maximum-tcp-connection-per-second-exceeded-as-server>0</maximum-tcp-connection-per-second-exceeded-
as-server>
        <maximum-udp-connection-exceeded-on-egress>0</maximum-udp-connection-exceeded-on-egress>
        <udp-packets-sent-as-is>0</udp-packets-sent-as-is>
        <out-pdu-preserve-on>0</out-pdu-preserve-on>
        <in-pdu-preserve-on>0</in-pdu-preserve-on>
        <total-ipv6-packets>12829</total-ipv6-packets>
        <flow-control-on>0</flow-control-on>
        <flow-control-off>0</flow-control-off>
        <active-udp-connections>0</active-udp-connections>
        <established-udp-connections>11643</established-udp-connections>
        <closed-udp-connections>11643</closed-udp-connections>
        <udp-route-lookup-fail>0</udp-route-lookup-fail>
        <udp-pdu-drops-due-to-pko-flow-control>0</udp-pdu-drops-due-to-pko-flow-control>
        <out-udp-pdus>0</out-udp-pdus>
        <out-udp-pdu-drops>0</out-udp-pdu-drops>
        <in-udp-pdus>20549</in-udp-pdus>
        <in-udp-tcl-lookup-fail-pdus>15</in-udp-tcl-lookup-fail-pdus>
        <in-udp-tcl-deny-pdus>1036</in-udp-tcl-deny-pdus>
        <in-total-udp-pdu-drops>1077</in-total-udp-pdu-drops>
        <in-udp-pdu-drops-due-to-stream-flow-control>0</in-udp-pdu-drops-due-to-stream-flow-control>
        <out-icmp-pdus>0</out-icmp-pdus>
```

```
        <out-icmp-pdu-drops>0</out-icmp-pdu-drops>
        <in-icmp-pdus>0</in-icmp-pdus>
        <in-icmp-tcl-lookup-fail-pdus>0</in-icmp-tcl-lookup-fail-pdus>
        <in-icmp-tcl-deny-pdus>0</in-icmp-tcl-deny-pdus>
        <in-total-icmp-pdu-drops>0</in-total-icmp-pdu-drops>
        <in-icmp-pdu-drops-due-to-stream-flow-control>0</in-icmp-pdu-drops-due-to-stream-flow-control>
        <out-as-is-ip-pdus>0</out-as-is-ip-pdus>
        <out-as-is-ip-pdu-drops>0</out-as-is-ip-pdu-drops>
        <in-as-is-ip-pdus>1764</in-as-is-ip-pdus>
        <in-as-is-ip-tcl-lookup-fail-pdus>12</in-as-is-ip-tcl-lookup-fail-pdus>
        <in-as-is-ip-tcl-deny-pdus>84</in-as-is-ip-tcl-deny-pdus>
        <in-total-as-is-ip-pdu-drops>96</in-total-as-is-ip-pdu-drops>
        <in-as-is-ip-pdu-drops-due-to-stream-flow-control>0</in-as-is-ip-pdu-drops-due-to-stream-flow-control>
        <in-error-ip-checksum>0</in-error-ip-checksum>
        <in-error-tcp-checksum>0</in-error-tcp-checksum>
        <in-error-mac>0</in-error-mac>
        <in-error-crc>0</in-error-crc>
        <in-error-parity>0</in-error-parity>
        <in-error-length>0</in-error-length>
        <out-udp-packets-64-bytes>0</out-udp-packets-64-bytes>
        <out-udp-packets-128-bytes>0</out-udp-packets-128-bytes>
        <out-udp-packets-256-bytes>0</out-udp-packets-256-bytes>
        <out-udp-packets-512-bytes>0</out-udp-packets-512-bytes>
        <out-udp-packets-1024-bytes>0</out-udp-packets-1024-bytes>
        <out-udp-packets-1500-bytes>0</out-udp-packets-1500-bytes>
        <out-udp-packets-3000-bytes>0</out-udp-packets-3000-bytes>
        <out-udp-packets-4500-bytes>0</out-udp-packets-4500-bytes>
        <out-udp-packets-6000-bytes>0</out-udp-packets-6000-bytes>
        <out-udp-packets-9000-bytes>0</out-udp-packets-9000-bytes>
        <in-udp-packets-64-bytes>6263</in-udp-packets-64-bytes>
        <in-udp-packets-128-bytes>9866</in-udp-packets-128-bytes>
        <in-udp-packets-256-bytes>880</in-udp-packets-256-bytes>
        <in-udp-packets-512-bytes>0</in-udp-packets-512-bytes>
        <in-udp-packets-1024-bytes>3540</in-udp-packets-1024-bytes>
        <in-udp-packets-1500-bytes>0</in-udp-packets-1500-bytes>
        <in-udp-packets-3000-bytes>0</in-udp-packets-3000-bytes>
        <in-udp-packets-4500-bytes>0</in-udp-packets-4500-bytes>
        <in-udp-packets-6000-bytes>0</in-udp-packets-6000-bytes>
        <in-udp-packets-9000-bytes>0</in-udp-packets-9000-bytes>
    </global-lan-statistics>
</Response>
```

brocade-extension-ip-route

This module is used to retrieve or configure IP route information on the specified switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-extension-ip-route.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-extension-ip-route
  +--rw brocade-extension-ip-route
     +--rw extension-ip-route* [name dp-id ip-address ip-prefix-length]
 {fibrechannel:fibrechannel_extension_platform}?
        +--rw name                brocade-interface-types:ip-extension-interface-type
        +--rw dp-id               extension:dp-id-type
        +--rw ip-address          union
        +--rw ip-prefix-length    brocade-interface-types:ip-prefix-length
        +--rw ip-gateway          brocade-interface-types:ip-gateway-type
        +--ro status-flags?       string
```

Examples

This section provides basic examples for using the brocade-extension-ip-route module.

Creating an IP Route

The following example uses POST to create an IP route to 3.0.0.0 using port 4/0 on the 2.2.2.3 gateway.

Structure

```
POST <base_URI>/running/brocade-extension-ip-route/extension-ip-route
```

URI

```
POST http://10.10.10.10/rest/running/brocade-extension-ip-route/extension-ip-route
```

Request Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<extension-ip-route>
   <name>4/0</name>
   <ip-address>3.0.0.0</ip-address>
   <ip-prefix-length>24</ip-prefix-length>
   <dp-id>0</dp-id>
   <ip-gateway>2.2.2.3</ip-gateway>
</extension-ip-route>
```

Response Body

When the resource is successfully deleted, the response contains an empty message body and a "201 Created" status in the first line of the headers.

Deleting an IP Route

The following example uses DELETE to delete the IP route to 3.0.0.0 from port 4/0.

Structure

```
DELETE <base_URI>/running/brocade-extension-ip-route/extension-ip-route
```

URI

```
DELETE http://10.10.10.10/rest/running/brocade-extension-ip-route/extension-ip-route
```

Request Body

```
<extension-ip-route>
   <name>4/0</name>
   <ip-address>3.0.0.0</ip-address>
   <ip-prefix-length>24</ip-prefix-length>
   <dp-id>0</dp-id>
</extension-ip-route>
```

Response Body

When the operation is successful, the response contains an empty message body and a "204 No Content" status in the first line of the headers.

Retrieving an IP Route

The following example uses GET to retrieve all the IP routes on the switch.

Structure

```
GET <base_URI>/running/brocade-extension-ip-route/extension-ip-route
```

URI

```
GET http://10.10.10.10/rest/running/brocade-extension-ip-route/extension-ip-route
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <extension-ip-route>
        <name>0/2</name>
        <dp-id>0</dp-id>
        <ip-address>7.9.0.0</ip-address>
        <ip-prefix-length>24</ip-prefix-length>
        <ip-gateway>12.18.20.1</ip-gateway>
        <status-flags>U G S </status-flags>
    </extension-ip-route>
    <extension-ip-route>
        <name>0/2</name>
        <dp-id>0</dp-id>
        <ip-address>12.18.20.0</ip-address>
```

```
        <ip-prefix-length>24</ip-prefix-length>
        <ip-gateway>*</ip-gateway>
        <status-flags>U C </status-flags>
    </extension-ip-route>
</Response>
```

brocade-extension-ipsec-policy

This module is used to retrieve or configure the IP security (IPsec) policy on the specified switch. IPsec policy creation supports creating a profile name using either a preshared format or a PKI format. The authentication-data attribute will contain the preshared key for a preshared profile and a key pair for a PKI profile.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-extension-ipsec-policy.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-extension-ipsec-policy
  +--rw brocade-extension-ipsec-policy
     +--rw extension-ipsec-policy* [policy-name] {fibrechannel:fibrechannel_extension_platform}?
        +--rw policy-name             string
        +--rw profile-name?           extension-ipsec-profile-name
        +--rw authentication-data?    authentication-data
        +--ro num-ike-sessions?       uint16
        +--rw restart-ike-sessions?   uint8
```

Examples

This section provides basic examples for using the brocade-extension-ipsec-policy module.

Creating an IPsec Policy Using a Preshared Profile

This example uses a POST request to create the IPsec policy “psk1” using a preshared profile.

Structure

```
POST <base_URI>/running/brocade-extension-ipsec-policy/extension-ipsec-policy
```

URI

```
POST https://10.10.10.10/rest/running/brocade-extension-ipsec-policy/extension-ipsec-policy
```

Request Body

```
<extension-ipsec-policy>
   <policy-name>psk1</policy-name>
   <profile-name>preshared</profile-name>
   <authentication-data>1234567891011121314</authentication-data>
</extension-ipsec-policy>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status in the headers.

Creating an IPsec Policy Using a PKI Profile

This example uses a POST request to create the IPsec policy “pki_1” using a PKI profile.

Structure

```
POST <base_URI>/running/brocade-extension-ipsec-policy/extension-ipsec-policy
```

URI

```
POST https://10.10.10.10/rest/running/brocade-extension-ipsec-policy/extension-ipsec-policy
```

Request Body

```
<extension-ipsec-policy>
   <policy-name>pki_1</policy-name>
   <profile-name>pki</profile-name>
   <authentication-data>pki-key-1</authentication-data>
</extension-ipsec-policy>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status in the headers.

Deleting an IPsec Policy

The following example uses a DELETE request to delete the IPsec policy named “psk1”.

Structure

```
DELETE <base_URI>/running/brocade-extension-ipsec-policy/extension-ipsec-policy/policy-name/policy_name
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-extension-ipsec-policy/extension-ipsec-policy/policy-name/psk1
```

Request Body

The request body is empty.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Retrieving a List of IPsec Policies

The following example uses a GET request to retrieve all the IPsec policies on the switch.

Structure

```
GET <base_URI>/running/brocade-extension-ipsec-policy/extension-ipsec-policy
```

URI

```
GET https://10.10.10.10/rest/running/brocade-extension-ipsec-policy/extension-ipsec-policy
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
  <extension-ipsec-policy>
    <policy-name>pki_policy</policy-name>
    <profile-name>pki</profile-name>
    <authentication-data>kprest</authentication-data>
    <num-ike-sessions>0</num-ike-sessions>
```

```
    <restart-ike-sessions>0</restart-ike-sessions>
  </extension-ipsec-policy>
  <extension-ipsec-policy>
    <policy-name>preshared_policy</policy-name>
    <profile-name>preshared</profile-name>
    <authentication-data/>
    <num-ike-sessions>0</num-ike-sessions>
    <restart-ike-sessions>0</restart-ike-sessions>
  </extension-ipsec-policy>
</Response>
```

brocade-extension-tunnel

This module is used to retrieve information on or configure Brocade Extension tunnels.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-extension-tunnel.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-extension-tunnel
  +--rw brocade-extension-tunnel
     +--rw extension-tunnel* [name] {fibrechannel:fibrechannel_extension_platform}?
     |  +--rw name                                        extension:ve-interface-type
     |  +--rw user-friendly-name?                         string
     |  +--ro local-wwn?                                  fibrechannel:wwn-type
     |  +--rw remote-wwn?                                 fibrechannel:wwn-type
     |  +--ro peer-wwn?                                   fibrechannel:wwn-type
     |  +--ro peer-product-name?                          fibrechannel:product-name-type
     |  x--rw admin-enabled?                              uint8
     |  +--rw admin-enabled-v2?                           boolean
     |  x--rw fast-write-enabled?                         uint8
     |  +--rw fast-write-enabled-v2?                      boolean
     |  x--rw tape-read?                                  uint8
     |  +--rw tape-read-v2?                               boolean
     |  x--rw tape-write?                                 uint8
     |  +--rw tape-write-v2?                              boolean
     |  +--ro tunnel-status?                              extension:operational-type
     |  +--ro tunnel-status-v2?                           extension:operational-status-type
     |  x--ro ipsec-enabled?                              uint8
     |  +--ro ipsec-enabled-v2?                           boolean
     |  +--rw ipsec-policy?                               string
     |  +--ro ha-operational-status?                      uint16
     |  x--rw ficon?                                      uint8
     |  +--rw ficon-v2?                                   boolean
     |  x--rw ficon-xrc-acceleration?                     uint8
     |  +--rw ficon-xrc-acceleration-v2?                  boolean
     |  x--rw ficon-tape-write-acceleration?              uint8
     |  +--rw ficon-tape-write-acceleration-v2?           boolean
     |  x--rw ficon-tape-read-acceleration?               uint8
     |  +--rw ficon-tape-read-acceleration-v2?            boolean
     |  x--rw ficon-tin-tir-emulation?                    uint8
     |  +--rw ficon-tin-tir-emulation-v2?                 boolean
     |  x--rw ficon-device-acknowledgement-emulation?     uint8
     |  +--rw ficon-device-acknowledgement-emulation-v2?  boolean
     |  x--rw ficon-read-block-id-emulation?              uint8
     |  +--rw ficon-read-block-id-emulation-v2?           boolean
     |  x--rw ficon-teradata-read-acceleration?           uint8
     |  +--rw ficon-teradata-read-acceleration-v2?        boolean
     |  x--rw ficon-teradata-write-acceleration?          uint8
     |  +--rw ficon-teradata-write-acceleration-v2?       boolean
     |  +--rw ficon-tape-write-max-pipe?                  uint32
     |  +--rw ficon-tape-read-max-pipe?                   uint32
```

```
|  +--rw ficon-tape-write-max-devices?               uint32
|  +--rw ficon-tape-read-max-devices?                uint32
|  +--rw ficon-tape-write-timer?                     uint32
|  +--rw ficon-tape-write-max-chain?                 uint32
|  +--rw ficon-oxid-base?                            uint32
|  x--rw ip-extension?                               uint16
|  +--rw ip-extension-v2?                            boolean
|  +--rw load-level?                                 extension:tunnel-load-level
|  +--ro active-load-level?                          extension:tunnel-load-level
|  +--ro peer-load-level?                            extension:tunnel-load-level
|  x--rw compression-tunnel?                         uint16
|  +--rw compression-tunnel-v2?                      extension:compression-type
|  +--rw compression-protocol
|  |  x--rw fc-compression?                          string
|  |  +--rw fc-compression-v2?                       extension:compression-type
|  |  x--rw ip-compression?                          string
|  |  +--rw ip-compression-v2?                       extension:compression-type
|  +--rw qos-ratio
|     +--rw distribution?                            string
|     +--rw distribution-value?                      string
|     +--rw fc-high-qos?                             string
|     +--rw fc-medium-qos?                           string
|     +--rw fc-low-qos?                              string
|     +--rw ip-high-qos?                             string
|     +--rw ip-medium-qos?                           string
|     +--rw ip-low-qos?                              string
+--ro extension-tunnel-statistics* [name]
|  +--ro name                                        extension:ve-interface-type
|  x--ro flow-status?                                uint16
|  +--ro flow-status-v2?                             boolean
|  +--ro operational-status?                         extension:operational-type
|  +--ro connection-count?                           yang:zero-based-counter32
|  +--ro duration?                                   uint32
|  +--ro uncompressed-bytes?                         yang:zero-based-counter64
|  +--ro compressed-bytes?                           yang:zero-based-counter64
|  +--ro local-hcl-in-progress?                      boolean
|  +--ro remote-hcl-in-progress?                     boolean
|  +--ro fc-ha-status?                               extension:operational-status-type
|  +--ro ip-ha-status?                               extension:operational-status-type
|  +--ro last-error?                                 uint32
+--rw extension-circuit* [name circuit-id]
                              {fibrechannel:fibrechannel_extension_platform}?
|  +--rw name                                        extension:ve-interface-type
|  +--rw circuit-id                                  uint16
|  +--rw local-ip-address?                           inet:ip-address
|  +--rw remote-ip-address?                          inet:ip-address
|  +--rw metric?                                     uint32
|  +--rw failover-group-id?                          uint32
|  x--rw admin-enabled?                              uint8
|  +--rw is-admin-enabled?                           boolean
|  x--ro admin-status?                               uint8
|  +--ro admin-status-v2?                            extension:circuit-admin-status-type
|  +--ro path-mtu-discovered?                        extension:discovered-mtu-type
```

```
|  +--ro ha-path-mtu-discovered?                    extension:discovered-mtu-type
|  +--rw minimum-communication-rate?                uint32
|  +--rw maximum-communication-rate?                uint32
|  +--rw keep-alive-timeout?                        uint32
|  +--ro vlan-id?                                   extension:vlan-id-type
|  +--ro ha-vlan-id?                                extension:vlan-id-type
|  +--rw l2-cos
|  |  +--rw priority-control?                       string
|  |  +--rw fc-priority-high?                       string
|  |  +--rw fc-priority-medium?                     string
|  |  +--rw fc-priority-low?                        string
|  |  +--rw ip-priority-high?                       string
|  |  +--rw ip-priority-medium?                     string
|  |  +--rw ip-priority-low?                        string
|  +--rw dscp
|  |  x--rw priority-control?                       string
|  |  x--rw fc-priority-high?                       string
|  |  x--rw fc-priority-medium?                     string
|  |  x--rw fc-priority-low?                        string
|  |  x--rw ip-priority-high?                       string
|  |  x--rw ip-priority-medium?                     string
|  |  x--rw ip-priority-low?                        string
|  |  +--rw priority-control-v2?                    inet:dscp
|  |  +--rw fc-priority-high-v2?                    inet:dscp
|  |  +--rw fc-priority-medium-v2?                  inet:dscp
|  |  +--rw fc-priority-low-v2?                     inet:dscp
|  |  +--rw ip-priority-high-v2?                    inet:dscp
|  |  +--rw ip-priority-medium-v2?                  inet:dscp
|  |  +--rw ip-priority-low-v2?                     inet:dscp
|  +--ro circuit-status?                            extension:operational-type
|  +--ro circuit-status-v2?                         extension:operational-status-type
|  +--rw local-ha-ip-address?                       inet:ip-address
|  +--rw remote-ha-ip-address?                      inet:ip-address
|  x--rw arl-algorithm-mode?                        uint32
|  +--rw arl-algorithm-mode-v2?                     extension:arl-algo-type
+--ro extension-circuit-statistics* [name circuit-id]
                             {fibrechannel:fibrechannel_extension_platform}?
|  +--ro name                                       extension:ve-interface-type
|  +--ro circuit-id                                 uint16
|  x--ro flow-status?                               uint16
|  +--ro flow-status-v2?                            boolean
|  +--ro operational-status?                        extension:operational-type
|  +--ro connection-count?                          yang:zero-based-counter32
|  +--ro duration?                                  uint32
|  +--ro out-packet-lost?                           yang:zero-based-counter64
|  +--ro out-packet-total?                          yang:zero-based-counter64
|  +--ro in-bytes?                                  yang:zero-based-counter64
|  +--ro out-bytes?                                 yang:zero-based-counter64
|  +--ro rtt?                                       yang:gauge32
|  +--ro time-generated?                            fibrechannel:time-generated-type
+--ro circuit-qos-statistics* [ve-port circuit-id ha-type priority]
                                {fibrechannel:fibrechannel_extension_platform}?
|  +--ro ve-port                                    extension:ve-interface-type
```

```
|  +--ro circuit-id                                  uint16
|  +--ro ha-type                                     extension:tunnel-ha-type
|  +--ro priority                                    extension:extension-qos-type
|  +--ro flow-status?                                boolean
|  +--ro operational-status?                         extension:operational-type
|  +--ro connection-count?                           yang:zero-based-counter32
|  +--ro duration?                                   uint32
|  +--ro in-bytes?                                   yang:zero-based-counter64
|  +--ro in-bytes-average?                           yang:gauge64
|  +--ro in-packets?                                 yang:zero-based-counter64
|  +--ro out-bytes?                                  yang:zero-based-counter64
|  +--ro out-bytes-average?                          yang:gauge64
|  +--ro out-packets?                                yang:zero-based-counter64
|  +--ro in-tcp-bytes?                               yang:zero-based-counter64
|  +--ro out-tcp-bytes?                              yang:zero-based-counter64
|  +--ro in-tcp-packets?                             yang:zero-based-counter64
|  +--ro out-tcp-packets?                            yang:zero-based-counter64
|  +--ro tcp-retransmits?                            yang:zero-based-counter64
|  +--ro rtt?                                        yang:zero-based-counter64
|  +--ro tcp-out-of-order-packets?                   yang:zero-based-counter64
|  +--ro tcp-slow-starts?                            yang:zero-based-counter64
o--ro wan-statistics* [ve-port circuit-id connection-id]
                                {fibrechannel:fibrechannel_extension_platform}?
|  o--ro ve-port                                     extension:ve-interface-type
|  o--ro circuit-id                                  uint16
|  o--ro connection-id                               uint32
|  o--ro priority?                                   extension:extension-qos-type
|  o--ro ha-type?                                    extension:tunnel-ha-type
|  o--ro source-port?                                inet:port-number
|  o--ro destination-port?                           inet:port-number
|  o--ro connection-mss?                             yang:zero-based-counter32
|  o--ro arl-minimum?                                yang:gauge32
|  o--ro arl-maximum?                                yang:gauge32
|  o--ro arl-current?                                yang:gauge32
|  o--ro arl-next-reset-algorithm?                   extension:arl-algo-type
|  o--ro out-bytes?                                  yang:zero-based-counter64
|  o--ro out-packets?                                yang:zero-based-counter64
|  o--ro rtt?                                        yang:gauge32
|  o--ro rtt-maximum?                                yang:gauge32
|  o--ro rtt-variance?                               yang:gauge32
|  o--ro rtt-variance-maximum?                       yang:gauge32
|  o--ro out-window-size?                            yang:gauge32
|  o--ro out-window-scale?                           uint32
|  o--ro slow-start-threshold?                       yang:gauge32
|  o--ro congestion-window?                          yang:gauge32
|  o--ro operation-mode?                             extension:tcp-operation-mode
|  o--ro out-queued-packets-next-sequence?           yang:zero-based-counter32
|  o--ro out-queued-packets-minimum-sequence?        yang:zero-based-counter32
|  o--ro out-queued-packets-maximum-sequence?        yang:zero-based-counter32
|  o--ro out-in-flight-packets?                      yang:gauge32
|  o--ro out-unacknowledged-packets-sequence?        yang:zero-based-counter32
|  o--ro retransmit-timeout?                         yang:zero-based-counter64
|  o--ro retransmits?                                yang:zero-based-counter64
```

```
|  o--ro maximum-retransmits?                          yang:zero-based-counter64
|  o--ro fast-retransmits?                             yang:zero-based-counter64
|  o--ro maximum-fast-retransmits?                     yang:zero-based-counter64
|  o--ro slow-retransmits?                             yang:zero-based-counter64
|  o--ro duplicate-acknowledgement?                    yang:zero-based-counter64
|  o--ro slow-starts?                                  yang:zero-based-counter64
|  o--ro in-bytes?                                     yang:zero-based-counter64
|  o--ro in-packets?                                   yang:zero-based-counter64
|  o--ro in-window-size?                               yang:zero-based-counter32
|  o--ro in-window-size-maximum?                       yang:zero-based-counter32
|  o--ro in-window-scale?                              uint32
|  o--ro in-queued-packets?                            yang:gauge32
|  o--ro in-queued-packets-next-sequence?              yang:zero-based-counter32
|  o--ro in-queued-packets-minimum-sequence?           yang:zero-based-counter32
|  o--ro in-queued-packets-maximum-sequence?           yang:zero-based-counter32
|  o--ro in-queued-out-of-order?                       yang:zero-based-counter32
|  o--ro in-queued-out-of-order-total?                 yang:zero-based-counter32
|  o--ro in-queued-out-of-order-maximum?               yang:zero-based-counter32
+--ro wan-statistics-v1* [ve-port circuit-id connection-id ha-type]
                            {fibrechannel:fibrechannel_extension_platform}?
   +--ro ve-port                                     extension:ve-interface-type
   +--ro circuit-id                                  uint16
   +--ro connection-id                               uint32
   +--ro ha-type                                     extension:tunnel-ha-type
   +--ro priority?                                   extension:extension-qos-type
   +--ro source-port?                                inet:port-number
   +--ro destination-port?                           inet:port-number
   +--ro connection-mss?                             yang:zero-based-counter32
   +--ro arl-minimum?                                yang:gauge32
   +--ro arl-maximum?                                yang:gauge32
   +--ro arl-current?                                yang:gauge32
   +--ro arl-next-reset-algorithm?                   extension:arl-algo-type
   +--ro out-bytes?                                  yang:zero-based-counter64
   +--ro out-packets?                                yang:zero-based-counter64
   +--ro rtt?                                        yang:gauge32
   +--ro rtt-maximum?                                yang:gauge32
   +--ro rtt-variance?                               yang:gauge32
   +--ro rtt-variance-maximum?                       yang:gauge32
   +--ro out-window-size?                            yang:gauge32
   +--ro out-window-scale?                           uint32
   +--ro slow-start-threshold?                       yang:gauge32
   +--ro congestion-window?                          yang:gauge32
   +--ro out-queued-packets-next-sequence?           yang:zero-based-counter32
   +--ro out-queued-packets-minimum-sequence?        yang:zero-based-counter32
   +--ro out-queued-packets-maximum-sequence?        yang:zero-based-counter32
   +--ro out-in-flight-packets?                      yang:gauge32
   +--ro out-unacknowledged-packets-sequence?        yang:zero-based-counter32
   +--ro fast-retransmits?                           yang:zero-based-counter64
   +--ro maximum-fast-retransmits?                   yang:zero-based-counter64
   +--ro slow-retransmits?                           yang:zero-based-counter64
   +--ro duplicate-acknowledgement?                  yang:zero-based-counter64
   +--ro slow-starts?                                yang:zero-based-counter64
   +--ro in-bytes?                                   yang:zero-based-counter64
```

```
+--ro in-packets?                              yang:zero-based-counter64
+--ro in-window-size?                          yang:zero-based-counter32
+--ro in-window-size-maximum?                  yang:zero-based-counter32
+--ro in-window-scale?                         uint32
+--ro in-queued-packets?                       yang:gauge32
+--ro in-queued-packets-next-sequence?         yang:zero-based-counter32
+--ro in-queued-packets-minimum-sequence?      yang:zero-based-counter32
+--ro in-queued-packets-maximum-sequence?      yang:zero-based-counter32
+--ro in-queued-out-of-order?                  yang:zero-based-counter32
+--ro in-queued-out-of-order-maximum?          yang:zero-based-counter32
```

Examples

This section provides basic examples for using the brocade-extension-tunnel module.

NOTE
The examples presented here illustrate only the key parameters required to create a tunnel or circuit. Multiple attributes can be provided in the data body of the request to set the values along with the key parameters.

Creating a Tunnel

This example uses a POST request to create a tunnel on port 4/17.

Structure

```
POST <base_URI>/running/brocade-extension-tunnel/extension-tunnel
```

URI

```
POST https://10.10.10.10/rest/running/brocade-extension-tunnel/extension-tunnel
```

Request Body

Refer to **Character Restrictions** in Using the Brocade FOS REST API section, when making FOS REST API requests.

```
<extension-tunnel>
   <name>4/17</name>
</extension-tunnel>
```

Response Body

When the operation is successful, the response contains an empty message body and a "201 Created" status in the headers.

Creating a Circuit on a Tunnel

This example uses a POST request to create circuit 0 on tunnel 4/17.

Structure

```
POST <base_URI>/running/brocade-extension-tunnel/extension-circuit
```

URI

```
POST https://10.10.10.10/rest/running/brocade-extension-tunnel/extension-circuit
```

Request body

```
<extension-circuit>
   <name>4/17</name>
```

```
    <circuit-id>0</circuit-id>
</extension-circuit>
```

Response Body

When the operation is successful, the response contains an empty message body and a "201 Created" status in the headers.

Deleting a Tunnel

This example uses a DELETE request to delete the tunnel and circuit running on port 4/17.

Structure

```
DELETE <base_URI>/running/brocade-extension-tunnel/extension-tunnel/name/4%2f17
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-extension-tunnel/extension-tunnel/name/4%2f17
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a "204 No Content" status in the headers.

Viewing a List of Circuit QoS Statistics

This example uses a GET request to retrieve a list of circuit QoS statistics.

Structure

```
GET base_URI/running/brocade-extension-tunnel/circuit-qos-statistics
```

URI

```
GET https://10.10.10.10/rest/running/brocade-extension-tunnel/circuit-qos-statistics
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <circuit-qos-statistics>
        <ve-port>8/16</ve-port>
        <circuit-id>1</circuit-id>
        <ha-type>main</ha-type>
        <priority>priority-control</priority>
        <duration>5053</duration>
        <connection-count>1</connection-count>
        <flow-status>false</flow-status>
        <operational-status>2</operational-status>
        <in-bytes>1474356</in-bytes>
```

```
        <in-bytes-average>81</in-bytes-average>
        <in-packets>4230</in-packets>
        <out-bytes>1231008</out-bytes>
        <out-bytes-average>176</out-bytes-average>
        <out-packets>4158</out-packets>
        <in-tcp-bytes>11851124</in-tcp-bytes>
        <out-tcp-bytes>14333824</out-tcp-bytes>
        <in-tcp-packets>99545</in-tcp-packets>
        <out-tcp-packets>99544</out-tcp-packets>
        <tcp-retransmits>0</tcp-retransmits>
        <rtt>1</rtt>
        <tcp-out-of-order-packets>0</tcp-out-of-order-packets>
        <tcp-slow-starts>0</tcp-slow-starts>
    </circuit-qos-statistics>
...
    <circuit-qos-statistics>
        <ve-port>8/17</ve-port>
        <circuit-id>4</circuit-id>
        <ha-type>main</ha-type>
        <priority>IP-priority-low</priority>
        <duration>0</duration>
        <connection-count>0</connection-count>
        <flow-status>true</flow-status>
        <operational-status>12</operational-status>
        <in-bytes>0</in-bytes>
        <in-bytes-average>0</in-bytes-average>
        <in-packets>0</in-packets>
        <out-bytes>0</out-bytes>
        <out-bytes-average>0</out-bytes-average>
        <out-packets>0</out-packets>
        <in-tcp-bytes>0</in-tcp-bytes>
        <out-tcp-bytes>0</out-tcp-bytes>
        <in-tcp-packets>0</in-tcp-packets>
        <out-tcp-packets>0</out-tcp-packets>
        <tcp-retransmits>0</tcp-retransmits>
        <rtt>0</rtt>
        <tcp-out-of-order-packets>0</tcp-out-of-order-packets>
        <tcp-slow-starts>0</tcp-slow-starts>
    </circuit-qos-statistics>
</Response>
```

Viewing the LAN Flow Statistics

This example uses a GET request to view the LAN flow statistics for the Extension switch or blade.

Structure

```
GET <base_URI>/running/running/brocade-extension/lan-flow-statistics/flow-index/flow_index/slot/slot_number/
dp-id/dp_id
```

URI

```
GET https://10.10.10.10/rest/running/brocade-extension/lan-flow-statistics/flow-index/7/slot/0/dp-id/0
```

Request Body

There is no request body.

Response Body

```
<?xml version="1.0"?>
<Response>
    <lan-flow-statistics>
        <flow-index>7</flow-index>
        <slot>0</slot>
        <dp-id>0</dp-id>
        <protocol>TCP</protocol>
        <source-port>61840</source-port>
        <destination-port>60771</destination-port>
        <source-ip-address>174.10.6.5</source-ip-address>
        <destination-ip-address>165.105.21.5</destination-ip-address>
        <lan-interface>0/6</lan-interface>
        <mapped-tunnel>
            <ve-port>0/12</ve-port>
            <qos>IP-priority-high</qos>
        </mapped-tunnel>
        <vlan-id>0</vlan-id>
        <vlan-priority>0</vlan-priority>
        <dscp>0</dscp>
        <hcl-flow>false</hcl-flow>
        <active-flow>true</active-flow>
        <start-time>2020-01-29T17:56:09-08:00</start-time>
        <rtt>0</rtt>
        <in-tcp-bytes>189636844</in-tcp-bytes>
        <in-tcp-packets>126487</in-tcp-packets>
        <out-tcp-bytes>1176216</out-tcp-bytes>
        <out-tcp-packets>29405</out-tcp-packets>
        <tcp-out-of-order-packets>0</tcp-out-of-order-packets>
        <tcp-retransmits>0</tcp-retransmits>
        <fast-retransmits>0</fast-retransmits>
        <slow-retransmits>0</slow-retransmits>
        <duplicate-acknowledgement>0</duplicate-acknowledgement>
        <zero-window-count>5355</zero-window-count>
        <zero-window-maximum-duration>17</zero-window-maximum-duration>
        <in-bytes-average>0</in-bytes-average>
        <out-bytes-average>0</out-bytes-average>
        <in-drops-lan-session-manager>0</in-drops-lan-session-manager>
        <out-drops-lan-session-manager>0</out-drops-lan-session-manager>
        <in-bytes-lan-session-manager>184577352</in-bytes-lan-session-manager>
        <out-bytes-lan-session-manager>4</out-bytes-lan-session-manager>
        <in-packets-lan-session-manager>0</in-packets-lan-session-manager>
        <out-packets-lan-session-manager>1</out-packets-lan-session-manager>
        <crc-errors>0</crc-errors>
        <local-host-mss>1460</local-host-mss>
        <remote-host-mss>1460</remote-host-mss>
```

```
</lan-flow-statistics>
```

brocade-fabric

This module is used to retrieve information on the switches in a fabric. If virtual fabrics are enabled, the request can include a query parameter (vf-id) for the desired virtual fabric. If no query parameter is specified and virtual fabrics are enabled, then the default virtual fabric name is returned.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-fabric.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-fabric
  +--ro brocade-fabric
     +--ro fabric-switch* [name] {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro name                          fibrechannel:wwn-type
     |  +--ro switch-user-friendly-name?    string
     |  +--ro chassis-wwn?                  fibrechannel:wwn-type
     |  +--ro chassis-user-friendly-name?   fibrechannel:chassis-user-friendly-name-type
     |  +--ro domain-id?                    fibrechannel:domain-id-type
     |  +--ro principal?                    uint8
     |  +--ro is-principal?                 boolean
     |  x--ro fcid?                         fibrechannel:fcid-type
     |  +--ro fcid-hex?                     fibrechannel:fcid-hex-string-type
     |  +--ro ip-address?                   inet:ipv4-address
     |  +--ro fcip-address?                 inet:ipv4-address
     |  +--ro ipv6-address?                 inet:ipv6-address
     |  +--ro firmware-version?             string
     |  x--ro path-count?                   yang:gauge32
     |  +--ro switch-model?                 string
     |  +--ro serial-number?                fru:serial-number-type
     +--ro access-gateway* [switch-wwn] {fibrechannel:fibrechannel_switch_platform}?
        +--ro switch-wwn            fibrechannel:wwn-type
        +--ro user-friendly-name?   string
        +--ro firmware-version?     string
        +--ro ip-addresses
        |  +--ro ip-address*   union
        +--ro is-edge-ag?           boolean
```

Examples

This section provides basic examples for using the brocade-fabric module.

Comparable Fabric OS CLI commands include `switchshow`, `fabricshow`, `wwn`, `chassisname`, and `ipaddrshow`. Refer to the *Brocade Fabric OS Command Reference Manual* for information and examples of these commands.

Retrieving Switch Information

This example uses a GET request to retrieve the switch information for a switch.

Structure

```
GET <base_URI>/running/brocade-fabric/fabric-switch
```

URI Request

```
GET https://10.10.10.10/rest/running/brocade-fabric/fabric-switch
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fabric-switch>
        <name>10:00:c4:f5:7c:4a:ac:6c</name>
        <chassis-wwn>10:00:c1:f1:1c:1a:ac:ab</chassis-wwn>
        <domain-id>43</domain-id>
        <fcid>16776235</fcid>
        <fcid-hex>0xfffc2b</fcid-hex>
        <switch-user-friendly-name>66-243_X6-4_Venator</switch-user-friendly-name>
        <chassis-user-friendly-name>BrocadeX6-4</chassis-user-friendly-name>
        <firmware-version>v9.0.0_bld95</firmware-version>
        <ip-address>10.38.66.243</ip-address>
        <fcip-address>0.0.0.0</fcip-address>
        <ipv6-address>2620:100:4:fa01:c6f5:7cff:fe4a:ac6c</ipv6-address>
        <principal>0</principal>
        <path-count>0</path-count>
    </fabric-switch>
    ...
    <fabric-switch>
        <name>10:00:c4:f5:7c:01:68:60</name>
        <chassis-wwn>10:00:c4:f5:7c:01:68:9f</chassis-wwn>
        <domain-id>238</domain-id>
        <fcid>16776430</fcid>
        <fcid-hex>0xfffcee</fcid-hex>
        <switch-user-friendly-name>66-238_G720_Cassian</switch-user-friendly-name>
        <chassis-user-friendly-name>G8000</chassis-user-friendly-name>
        <firmware-version>v9.0.0_bld95</firmware-version>
        <ip-address>10.38.66.238</ip-address>
        <fcip-address>0.0.0.0</fcip-address>
        <ipv6-address>2620:100:4:fa01:c6f5:7cff:fe01:6860</ipv6-address>
        <principal>0</principal>
        <path-count>3</path-count>
    </fabric-switch>
</Response>
```

Retrieving Switch Information From a Virtual Fabric

This example uses a GET request to retrieve the switch information for a switch in the virtual fabric with an VFID of 10.

Structure

```
GET <base_URI>/running/brocade-fabric/fabric-switch?vf-id=<vf-id#>
```

URI Request

```
GET https://10.10.10.10/rest/running/brocade-fabric/fabric-switch?vf-id=10
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
   <fabric-switch>
      <name>10:10:14:15:1c:9e:3b:44</name>
      <chassis-wwn>10:10:14:15:1c:9e:3c:aa</chassis-wwn>
      <domain-id>5</domain-id>
      <fcid>16776100</fcid>
      <switch-user-friendly-name>G610_21</switch-user-friendly-name>
      <chassis-user-friendly-name>BrocadeG610</chassis-user-friendly-name>
      <firmware-version>v820</firmware-version>
      <vf-id>1</vf-id>
      <ip-address>10.10.20.21</ip-address>
      <fcip-address>0.0.0.0</fcip-address>
      <ipv6-address>::</ipv6-address>
      <principal>1</principal>
      <path-count>2</path-count>
    </fabric-switch>
    <fabric-switch>
      <name>10:10:14:15:1c:a2:1f:40</name>
      <chassis-wwn>10:10:14:15:1c:a2:1f:af</chassis-wwn>
      <domain-id>1</domain-id>
      <fcid>16776101</fcid>
      <switch-user-friendly-name>G610_22</switch-user-friendly-name>
      <chassis-user-friendly-name>BrocadeG610</chassis-user-friendly-name>
      <firmware-version>v820</firmware-version>
      <vf-id>1</vf-id>
      <ip-address>10.10.20.1</ip-address>
      <fcip-address>0.0.0.0</fcip-address>
      <ipv6-address>::</ipv6-address>
      <principal>0</principal>
      <path-count>1</path-count>
    </fabric-switch>
</Response>
```

Viewing Attached Access Gateway Switches

This example uses a GET request to view a list of the Access Gateway switches attached to the fabric in non-Access Gateway mode.

Structure

```
GET <base_URI>/running/brocade-fabric/access-gateway/switch-wwn/{switch-wwn}/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fabric/access-gateway/switch-wwn/10:00:10:10:10:1e:10:40
```

Request Body

The request body is not required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <access-gateway>
        <switch-wwn>10:00:10:10:10:1e:10:40</switch-wwn>
        <user-friendly-name>66-62_G620_Wedge</user-friendly-name>
        <ip-addresses>
            <ip-address>10.38.66.62</ip-address>
            <ip-address>2020:100:4:ff01:c2f2:7cff:fed2:9be3</ip-address>
        </ip-addresses>
        <firmware-version>v9.0.0</firmware-version>
        <is-edge-ag>false</is-edge-ag>
    </access-gateway>
</Response>
```

brocade-fabric-traffic-controller

The Fabric Traffic Controller (FTC) acts as a central point of processing to translate Fabric Performance Impacts (FPI) into notifications and signaling to end devices throughout the Fabric. The FTC uses signals and notifications to communicate the type and relative severity of impacting behaviors to the affected members of the Fabric. The FTC records information regarding diagnostic Extended Link Service (ELS) registration, FPI conditions, and related statistics; storing the information per device (per N_Port).

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-fabric-traffic-controller.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-fabric-traffic-controller
  +--ro brocade-fabric-traffic-controller
     +--ro fabric-traffic-controller-device* [n-port-id] {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro n-port-id                          fibrechannel:fcid-hex-string-type
     |  +--ro n-port-wwn?                        fibrechannel:wwn-type
     |  +--ro f-port-wwn?                        fibrechannel:wwn-type
     |  +--ro port-index?                        fibrechannel:user-port-number-type
     |  +--ro rdf-registrations
     |  |  +--ro rdf-registration*   els-descriptor-type
     |  +--ro congestion-states
     |  |  +--ro congestion-state*   congestion-state-type
     |  +--ro fpin-send-statistics
     |  |  +--ro peer-congestion-count?   yang:counter32
     |  |  +--ro congestion-count?        yang:counter32
     |  |  +--ro link-integrity-count?    yang:counter32
     |  |  +--ro delivery-count?          yang:counter32
     |  +--ro fpin-receive-statistics
     |  |  +--ro peer-congestion-count?   yang:counter32
     |  |  +--ro congestion-count?        yang:counter32
     |  |  +--ro link-integrity-count?    yang:counter32
     |  |  +--ro delivery-count?          yang:counter32
     |  +--ro edc-owner?                         fibrechannel:fcid-hex-string-type
     |  +--ro n-port-diagnostic-capabilities
     |  |  +--ro tx-signal-capability?           congestion-detection-signal-type
     |  |  +--ro tx-signal-cycle?                congestion-detection-signal-cycles-type
     |  |  +--ro tx-signal-scale?                congestion-detection-signal-scale-type
     |  |  +--ro rx-signal-capability?           congestion-detection-signal-type
     |  |  +--ro rx-signal-cycle?                congestion-detection-signal-cycles-type
     |  |  +--ro rx-signal-scale?                congestion-detection-signal-scale-type
     |  |  +--ro degrade-activate-threshold?     uint32
     |  |  +--ro degrade-deactivate-threshold?   uint32
     |  |  +--ro fec-degrade-interval?           uint32
     |  +--ro f-port-diagnostic-capabilities
     |  |  +--ro tx-signal-capability?           congestion-detection-signal-type
     |  |  +--ro tx-signal-cycle?                congestion-detection-signal-cycles-type
     |  |  +--ro tx-signal-scale?                congestion-detection-signal-scale-type
     |  |  +--ro rx-signal-capability?           congestion-detection-signal-type
     |  |  +--ro rx-signal-cycle?                congestion-detection-signal-cycles-type
     |  |  +--ro rx-signal-scale?                congestion-detection-signal-scale-type
```

```
|  |  +--ro degrade-activate-threshold?     uint32
|  |  +--ro degrade-deactivate-threshold?   uint32
|  |  +--ro fec-degrade-interval?           uint32
|  +--ro applied-signaling-capabilities
|     +--ro tx-signal-capability?   congestion-detection-signal-type
|     +--ro tx-signal-cycle?        congestion-detection-signal-cycles-type
|     +--ro tx-signal-scale?        congestion-detection-signal-scale-type
|     +--ro rx-signal-capability?   congestion-detection-signal-type
|     +--ro rx-signal-cycle?        congestion-detection-signal-cycles-type
|     +--ro rx-signal-scale?        congestion-detection-signal-scale-type
+--ro fabric-traffic-controller-ag-f-port* [port-index] {fibrechannel:access_gateway_platform}?
|  +--ro port-index                       fibrechannel:user-port-number-type
|  +--ro ag-mapped-n-port-index?          fibrechannel:user-port-number-type
|  +--ro congestion-states
|  |  +--ro congestion-state*   congestion-state-type
|  +--ro edc-owner?                       fibrechannel:fcid-hex-string-type
|  +--ro n-port-diagnostic-capabilities
|  |  +--ro tx-signal-capability?          congestion-detection-signal-type
|  |  +--ro tx-signal-cycle?               congestion-detection-signal-cycles-type
|  |  +--ro tx-signal-scale?               congestion-detection-signal-scale-type
|  |  +--ro rx-signal-capability?          congestion-detection-signal-type
|  |  +--ro rx-signal-cycle?               congestion-detection-signal-cycles-type
|  |  +--ro rx-signal-scale?               congestion-detection-signal-scale-type
|  |  +--ro degrade-activate-threshold?    uint32
|  |  +--ro degrade-deactivate-threshold?  uint32
|  |  +--ro fec-degrade-interval?          uint32
|  +--ro f-port-diagnostic-capabilities
|  |  +--ro tx-signal-capability?          congestion-detection-signal-type
|  |  +--ro tx-signal-cycle?               congestion-detection-signal-cycles-type
|  |  +--ro tx-signal-scale?               congestion-detection-signal-scale-type
|  |  +--ro rx-signal-capability?          congestion-detection-signal-type
|  |  +--ro rx-signal-cycle?               congestion-detection-signal-cycles-type
|  |  +--ro rx-signal-scale?               congestion-detection-signal-scale-type
|  |  +--ro degrade-activate-threshold?    uint32
|  |  +--ro degrade-deactivate-threshold?  uint32
|  |  +--ro fec-degrade-interval?          uint32
|  +--ro applied-signaling-capabilities
|     +--ro tx-signal-capability?   congestion-detection-signal-type
|     +--ro tx-signal-cycle?        congestion-detection-signal-cycles-type
|     +--ro tx-signal-scale?        congestion-detection-signal-scale-type
|     +--ro rx-signal-capability?   congestion-detection-signal-type
|     +--ro rx-signal-cycle?        congestion-detection-signal-cycles-type
|     +--ro rx-signal-scale?        congestion-detection-signal-scale-type
+--ro fabric-traffic-controller-ag-n-port* [port-index] {fibrechannel:access_gateway_platform}?
   +--ro port-index                                  fibrechannel:user-port-number-type
   +--ro port-id?                                    fibrechannel:fcid-hex-string-type
   +--ro ag-n-port-notification-capabilities
   |  +--ro capability*   notification-capabilities-type
   +--ro fabric-neighbor-notification-capabilities
      +--ro capability*   notification-capabilities-type
```

Examples

This section provides basic examples for using the brocade-fabric-traffic-controller module.

Retrieving Fabric Traffic Controller Information

This example uses a GET request to retrieve FTC diagnostic information for the device to which the request is sent.

Structure

```
GET <base_URI>/running/brocade-fabric-traffic-controller/fabric-traffic-controller-device
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fabric-traffic-controller/fabric-traffic-controller-device
```

Request Body

The request body is empty.

Response Body

```
<Response>
   <fabric-traffic-controller-device>
       <n-port-id>0x030300</n-port-id>
       <n-port-wwn>30:10:50:eb:1a:bb:6c:fd</n-port-wwn>
       <f-port-wwn>20:02:00:27:f8:f0:3a:76</f-port-wwn>
       <port-index>2</port-index>
       <rdf-registrations>
           <rdf-registration>fpin-congestion</rdf-registration>
           <rdf-registration>fpin-peer-congestion</rdf-registration>
           <rdf-registration>fpin-link-integrity</rdf-registration>
           <rdf-registration>fpin-delivery</rdf-registration>
       </rdf-registrations>
       <congestion-states>
           <congestion-state>none</congestion-state>
       </congestion-states>
       <edc-owner>0x030300</edc-owner>
       <fpin-send-statistics>
           <peer-congestion-count>0</peer-congestion-count>
           <congestion-count>0</congestion-count>
           <link-integrity-count>0</link-integrity-count>
           <delivery-count>0</delivery-count>
       </fpin-send-statistics>
       <fpin-receive-statistics>
           <peer-congestion-count>0</peer-congestion-count>
           <congestion-count>0</congestion-count>
           <link-integrity-count>0</link-integrity-count>
           <delivery-count>0</delivery-count>
       </fpin-receive-statistics>
       <n-port-diagnostic-capabilities>
           <tx-signal-capability>warning</tx-signal-capability>
           <tx-signal-cycle>100</tx-signal-cycle>
           <tx-signal-scale>milliseconds</tx-signal-scale>
           <rx-signal-capability>warning</rx-signal-capability>
```

```
            <rx-signal-cycle>100</rx-signal-cycle>
            <rx-signal-scale>milliseconds</rx-signal-scale>
            <degrade-activate-threshold>20</degrade-activate-threshold>
            <degrade-deactivate-threshold>10</degrade-deactivate-threshold>
            <fec-degrade-interval>2</fec-degrade-interval>
        </n-port-diagnostic-capabilities>
        <f-port-diagnostic-capabilities>
            <tx-signal-capability>none</tx-signal-capability>
            <tx-signal-cycle>0</tx-signal-cycle>
            <tx-signal-scale>none</tx-signal-scale>
            <rx-signal-capability>none</rx-signal-capability>
            <rx-signal-cycle>0</rx-signal-cycle>
            <rx-signal-scale>none</rx-signal-scale>
            <degrade-activate-threshold>0</degrade-activate-threshold>
            <degrade-deactivate-threshold>0</degrade-deactivate-threshold>
            <fec-degrade-interval>0</fec-degrade-interval>
        </f-port-diagnostic-capabilities>
        <applied-signaling-capabilities>
            <tx-signal-capability>none</tx-signal-capability>
            <tx-signal-cycle>0</tx-signal-cycle>
            <tx-signal-scale>none</tx-signal-scale>
            <rx-signal-capability>none</rx-signal-capability>
            <rx-signal-cycle>0</rx-signal-cycle>
            <rx-signal-scale>none</rx-signal-scale>
        </applied-signaling-capabilities>
    </fabric-traffic-controller-device>
</Response>
```

Retrieving Fabric Traffic Controller Information for a Specific N_Port

This example uses a GET request to retrieve FTC diagnostic information for the 0x030300 N_Port ID on the device to which the request is sent.

Structure

```
GET <base_URI>/running/brocade-fabric-traffic-controller/fabric-traffic-controller-device/n-port-id/
<n_port_id>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fabric-traffic-controller/fabric-traffic-controller-device/n-
port-id/0x030300
```

Request Body

The request body is empty.

Response Body

```
<Response>
   <fabric-traffic-controller-device>
       <n-port-id>0x030300</n-port-id>
       <n-port-wwn>30:10:50:eb:1a:bb:6c:fd</n-port-wwn>
       <f-port-wwn>20:02:00:27:f8:f0:3a:76</f-port-wwn>
```

```
<port-index>2</port-index>
<rdf-registrations>
    <rdf-registration>fpin-congestion</rdf-registration>
    <rdf-registration>fpin-peer-congestion</rdf-registration>
    <rdf-registration>fpin-link-integrity</rdf-registration>
    <rdf-registration>fpin-delivery</rdf-registration>
</rdf-registrations>
<congestion-states>
    <congestion-state>none</congestion-state>
</congestion-states>
<edc-owner>0x030300</edc-owner>
<fpin-send-statistics>
    <peer-congestion-count>0</peer-congestion-count>
    <congestion-count>0</congestion-count>
    <link-integrity-count>0</link-integrity-count>
    <delivery-count>0</delivery-count>
</fpin-send-statistics>
<fpin-receive-statistics>
    <peer-congestion-count>0</peer-congestion-count>
    <congestion-count>0</congestion-count>
    <link-integrity-count>0</link-integrity-count>
    <delivery-count>0</delivery-count>
</fpin-receive-statistics>
<n-port-diagnostic-capabilities>
    <tx-signal-capability>warning</tx-signal-capability>
    <tx-signal-cycle>100</tx-signal-cycle>
    <tx-signal-scale>milliseconds</tx-signal-scale>
    <rx-signal-capability>warning</rx-signal-capability>
    <rx-signal-cycle>100</rx-signal-cycle>
    <rx-signal-scale>milliseconds</rx-signal-scale>
    <degrade-activate-threshold>20</degrade-activate-threshold>
    <degrade-deactivate-threshold>10</degrade-deactivate-threshold>
    <fec-degrade-interval>2</fec-degrade-interval>
</n-port-diagnostic-capabilities>
<f-port-diagnostic-capabilities>
    <tx-signal-capability>none</tx-signal-capability>
    <tx-signal-cycle>0</tx-signal-cycle>
    <tx-signal-scale>none</tx-signal-scale>
    <rx-signal-capability>none</rx-signal-capability>
    <rx-signal-cycle>0</rx-signal-cycle>
    <rx-signal-scale>none</rx-signal-scale>
    <degrade-activate-threshold>0</degrade-activate-threshold>
    <degrade-deactivate-threshold>0</degrade-deactivate-threshold>
    <fec-degrade-interval>0</fec-degrade-interval>
</f-port-diagnostic-capabilities>
<applied-signaling-capabilities>
    <tx-signal-capability>none</tx-signal-capability>
    <tx-signal-cycle>0</tx-signal-cycle>
    <tx-signal-scale>none</tx-signal-scale>
    <rx-signal-capability>none</rx-signal-capability>
    <rx-signal-cycle>0</rx-signal-cycle>
    <rx-signal-scale>none</rx-signal-scale>
</applied-signaling-capabilities>
```

```
    </fabric-traffic-controller-device>
</Response>
```

brocade-fdmi

This module retrieves Fabric Device Management Interface (FDMI) information for the specified switch. FDMI enables discovery of devices such as Fibre Channel host bus adapters (HBAs). If virtual fabrics are enabled, the request can include a query parameter (vf-id) for the desired virtual fabric. If no query parameter is specified and virtual fabrics are enabled, the information for the switch in the default virtual fabric is returned.

For a list of changes to this module by release, refer to the individual module YANG file.

It assumes a knowledge of FDMI as performed in Fabric OS. For information on that topic, refer to the `fdmiShow` command in the *Brocade Fabric OS Command Reference Manual*.

NOTE
The brocade-fdmi module is supported in Fabric OS 8.2.0a and later.

Module Tree

This is the tree view of the module from the brocade-fdmi.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Typesfor data type descriptions.

```
module: brocade-fdmi
  +--ro brocade-fdmi
     +--ro hba* [hba-id]
     |  +--ro hba-id                 fibrechannel:wwn-type
     |  +--ro domain-id?             fibrechannel:domain-id-type
     |  +--ro manufacturer?          string
     |  +--ro serial-number?         string
     |  +--ro model?                 string
     |  +--ro model-description?     string
     |  +--ro node-name?             fibrechannel:wwn-type
     |  +--ro node-symbolic-name?    string
     |  +--ro hardware-version?      string
     |  +--ro driver-version?        string
     |  +--ro option-rom-version?    string
     |  +--ro firmware-version?      string
     |  +--ro os-name-and-version?   string
     |  +--ro max-ct-payload?        uint32
     |  +--ro vendor-id?             string
     |  +--ro vendor-specific-info?  fibrechannel:brocade-hex-string-type
     |  +--ro number-of-ports?       uint32
     |  +--ro fabric-name?           fibrechannel:wwn-type
     |  +--ro boot-bios-version?     string
     |  +--ro boot-bios-enabled?     uint8
     |  +--ro boot-bios-enabled-v2?  boolean
     |  +--ro hba-port-list
     |     +--ro wwn*   fibrechannel:wwn-type
     +--ro port* [port-name]
        +--ro port-name                         fibrechannel:wwn-type
        +--ro hba-id?                           fibrechannel:wwn-type
        +--ro domain-id?                        fibrechannel:domain-id-type
        +--ro port-symbolic-name?               string
        +--ro port-id?                          fibrechannel:fcid-hex-string-type
        +--ro port-type?                        fibrechannel:port-type-string-type
        +--ro supported-class-of-service?       fibrechannel:class-of-service-type
        +--ro supported-fc4-type?               fibrechannel:fc4-type-type
```

```
+--ro active-fc4-type?                  fibrechannel:fc4-type-type
x--ro supported-speed?                  fibrechannel:speed-type
+--ro supported-protocol-speeds
|  +--ro supported-protocol-speed*   fibrechannel:protocol-speed-type
x--ro current-port-speed?               fibrechannel:speed-type
+--ro current-protocol-speed?           fibrechannel:protocol-speed-type
+--ro maximum-frame-size?               uint32
+--ro os-device-name?                   string
+--ro host-name?                        string
+--ro node-name?                        fibrechannel:wwn-type
+--ro fabric-name?                      fibrechannel:wwn-type
+--ro port-state?                       fibrechannel:brocade-hex-string-type
+--ro number-of-discovered-ports?       uint32
+--ro vsa-service-category?             string
+--ro vsa-guid?                         string
+--ro vsa-version?                      string
+--ro vsa-product-name?                 string
+--ro vsa-port-info?                    string
+--ro vsa-qos-supported?                string
+--ro vsa-security?                     string
+--ro vsa-storage-array-family?         string
+--ro vsa-storage-array-name?           string
+--ro vsa-storage-array-system-model?   string
+--ro vsa-storage-array-os?             string
+--ro vsa-storage-array-node-count?     uint32
+--ro vsa-storage-array-nodes
|  +--ro nodes*   string
+--ro vsa-connected-ports
|  +--ro wwns*   fibrechannel:wwn-type
+--ro vsa-end-to-end-version?           string
```

Examples

This section provides a basic example for using the brocade-fdmi module.

Retrieving HBA Port Attributes

This example uses a GET request to retrieve port attributes.

Structure

GET *<base_URI>*/running/brocade-fdmi/port

URI

```
GET https://10.10.10.10/rest/running/brocade-fdmi/port
```

Request Body

The request body is empty.

Response Body

```
<?xml version="1.0"?>
<Response>
    <port>
```

```
        <port-name>21:00:00:0e:1e:07:03:48</port-name>
        <hba-id>21:00:00:0e:1e:07:03:48</hba-id>
        <domain-id>43</domain-id>
        <port-symbolic-name>QLE2670 FW:v6.02.00 DVR:v9.2.9.23</port-symbolic-name>
        <port-id>0x2baa00</port-id>
        <port-type>n-port</port-type>
        <supported-class-of-service>class-3</supported-class-of-service>
        <supported-fc4-type>FCP</supported-fc4-type>
        <active-fc4-type>FCP</active-fc4-type>
        <supported-speed>speed-4-gfc speed-8-gfc speed-16-gfc</supported-speed>
        <current-port-speed>speed-16-gfc</current-port-speed>
        <maximum-frame-size>2048</maximum-frame-size>
        <os-device-name>QLogic Adapter</os-device-name>
        <host-name>W16-BRM68164</host-name>
        <node-name>20:00:00:0e:1e:07:01:41</node-name>
        <fabric-name>10:00:c1:f5:1c:1a:ac:1c</fabric-name>
        <port-state>0x2</port-state>
        <number-of-discovered-ports>0</number-of-discovered-ports>
    </port>
...
    <port>
        <port-name>10:00:00:10:1b:1f:1b:c1</port-name>
        <hba-id>10:00:00:10:1b:1f:1b:c1</hba-id>
        <domain-id>43</domain-id>
        <port-symbolic-name>Emulex PPN-10:00:00:10:1B:1F:1B:C1</port-symbolic-name>
        <port-id>0x2bad00</port-id>
        <port-type>n-port</port-type>
        <supported-class-of-service>class-3</supported-class-of-service>
        <supported-fc4-type>0x01 FCP FC-CT FC-NVMe</supported-fc4-type>
        <active-fc4-type>0x01 FCP FC-CT</active-fc4-type>
        <supported-speed>speed-8-gfc speed-16-gfc speed-32-gfc</supported-speed>
        <current-port-speed>speed-32-gfc</current-port-speed>
        <maximum-frame-size>2048</maximum-frame-size>
        <host-name>W16-68166</host-name>
        <node-name>20:00:00:10:1b:1f:1b:c1</node-name>
        <fabric-name>20:ad:c1:f1:1c:1a:ac:1c</fabric-name>
        <port-state>0x2</port-state>
        <number-of-discovered-ports>3</number-of-discovered-ports>
        <vsa-end-to-end-version>ELXE2EM:0001</vsa-end-to-end-version>
    </port>
</Response>
```

brocade-fibrechannel-configuration

This module enables you to configure a switch or director.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of this module from the `brocade-fibrechannel-configuration.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types

```
module: brocade-fibrechannel-configuration
  +--rw brocade-fibrechannel-configuration
     +--rw switch-configuration
     |  +--rw trunk-enabled?      boolean
     |  +--rw wwn-port-id-mode?   boolean {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw edge-hold-time?     uint16
     |  x--rw area-mode?          uint8 {fibrechannel:fibrechannel_chassis_platform}?
     |  +--rw area-mode-v2?       enumeration
                                     {fibrechannel:fibrechannel_chassis_platform}?
     |  +--rw xisl-enabled?       boolean
     +--rw f-port-login-settings
     |  +--rw max-logins?                  uint16
     |  +--rw max-flogi-rate-per-switch?   uint16
     |  +--rw stage-interval?              uint16
     |  +--rw free-fdisc?                  uint16
     |  x--rw enforce-login?               uint16
     |  +--rw enforce-login-string?        enumeration
     |  +--rw max-flogi-rate-per-port?     uint16
     +--rw port-configuration
     |  +--rw portname-mode?              string
     |  +--rw dynamic-portname-format?    string {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw dynamic-d-port-enabled?     boolean
     |  +--rw on-demand-d-port-enabled?   boolean {fibrechannel:fibrechannel_switch_platform}?
     +--rw zone-configuration {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw node-name-zoning-enabled?   boolean {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw fabric-lock-timeout?        uint16
     +--rw fabric
     |  +--rw insistent-domain-id-enabled?        boolean {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw principal-selection-enabled?        boolean {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw principal-priority?                 fibrechannel:brocade-hex-string-type
  {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw preserved-domain-id-mode-enabled?   boolean {fibrechannel:fibrechannel_switch_platform}?
     +--rw chassis-config-settings
        +--rw firmware-synchronization-enabled?   boolean
        +--rw http-session-ttl?                   uint32
        o--rw ezserver-enabled?                   boolean {fibrechannel:fibrechannel_switch_platform}?
        +--rw cs-ctl-mode?                        enumeration
        +--rw sddq-limit?                         uint32
        +--rw vtap-qos-compatibility-mode?        boolean
        +--rw secure-mode-enabled?                boolean
        +--rw file-suffix-enabled?                boolean
        +--rw custom-index?                       uint32
```

for data type descriptions.

Examples

This section provides basic examples for using the brocade-fibrechannel-configuration module.

Configuring the Maximum Number of Logins Allowed on a Port

The following example uses the PATCH request to configure the maximum number of logins allowed per second on a port.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-configuration/f-port-login-settings
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-configuration/f-port-login-settings
```

Request Body

```
<f-port-login-settings>
   <max-flogi-rate-per-port>100</max-flogi-rate-per-port>
</f-port-login-settings>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Viewing the Port Configuration

The following example uses the GET request to view the existing port configuration.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-configuration/port-configuration
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-configuration/port-configuration
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <port-configuration>
        <portname-mode>default</portname-mode>
        <dynamic-portname-format>S.T.I</dynamic-portname-format>
        <dynamic-d-port-enabled>true</dynamic-d-port-enabled>
        <on-demand-d-port-enabled>false</on-demand-d-port-enabled>
    </port-configuration>
</Response>
```

Configuring the Dynamic Port Name Format

The following example uses the PATCH request to configure the dynamic port name format as S.T.I.A, which represents *switch_name*.*port_type*.*port_index*.*alias_name*. The format of a dynamic port name is composed of fields that are mapped using the following characters:

S = Switch name

T = Port type

I = Port index

C = Slot number/port number (only applicable on a chassis)

A = Alias name

F = FDMI host name

R = Remote switch name

Multiple unique fields may be specified, separated by hyphens (-), periods (.), or underscores (_). Fields may not be repeated, and field separators must be the same. The switch must be an FC switch in Native mode.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-configuration/port-configuration
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-configuration/port-configuration
```

Request Body

```
<port-configuration>
  <dynamic-portname-format>S.T.I.A</dynamic-portname-format>
</port-configuration>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content" status appears in the header.

Enabling Node Name Checking for Zoning

The following example uses the PATCH request to enable using the node WWN when specifying nodes for zoning.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-configuration/zone-configuration
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-configuration/zone-configuration
```

Request Body

```
<zone-configuration>
    <node-name-zoning-enabled>true</node-name-zoning-enabled>
</zone-configuration>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content" status appears in the header.

Modifying the Fabric Lock Timeout

This example uses a PATCH request to modify the fabric lock timeout duration to 6 minutes.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-configuration/zone-configuration/fabric-lock-timeout/6
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-configuration/zone-configuration/fabric-lock-timeout/6
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content" status appears in the header.

Enabling Insistent Domain ID on a Device

The following example uses the PATCH request to enable Insistent Domain ID on a device.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-configuration/fabric
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-configuration/fabric
```

Request Body

```
<fabric>
  <insistent-domain-id-enabled>true</insistent-domain-id-enabled>
</fabric>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content" status appears in the header.

Configuring the Fabric

This example uses a PATCH request to configure the fabric data for a switch.

Structure

```
PATCH base_URI/running/brocade-fibrechannel-configuration/fabric
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-configuration/fabric
```

Request Body

```
<fabric>
    <insistent-domain-id-enabled>false</insistent-domain-id-enabled>
    <principal-selection-enabled>false</principal-selection-enabled>
    <principal-priority>0x00</principal-priority>
</fabric>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status in the headers.

Configuring the Chassis Settings

This example uses a PATCH request to configure the chassis settings.

Structure

```
PATCH base_URI/running/brocade-fibrechannel-configuration/chassis-config-settings
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-configuration/chassis-config-settings
```

Request Body

```
 <chassis-config-settings>
     <firmware-synchronization-enabled>false</firmware-synchronization-enabled>
     <http-session-ttl>7200</http-session-ttl>
     <ezserver-enabled>true</ezserver-enabled>
 </chassis-config-settings>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status in the headers.

brocade-fibrechannel-diagnostics

This module provides support for ClearLink diagnostic port configuration and diagnostic port results. It expects that the switch is configured with an actual online diagnostics port (D_Port).

For a list of changes to this module by release, refer to the individual module YANG file.

For information on diagnostics, refer to the *Brocade Fabric OS Troubleshooting and Diagnostics Reference Manual*.

Module Tree

This is the tree view of the module from the `brocade-fibrechannel-diagnostics.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-fibrechannel-diagnostics
  +--rw brocade-fibrechannel-diagnostics
     +--rw fibrechannel-diagnostics* [name]
        +--rw name                              fibrechannel:slot-port-name-type
        +--rw diagnostic-control?               uint32
        +--ro mode?                             enumeration
        +--ro state?                            string
        +--ro error-message?                    string
        +--ro distance?                         int32
        +--ro remote-switch-wwn?                fibrechannel:wwn-type
        +--ro remote-port-index?                fibrechannel:user-port-number-type
        o--ro electrical-loopback-test
        |  o--ro result?          result-type
        |  o--ro comments?        comments-type
        |  o--ro start-time?      fibrechannel:time-24hr-extended-type
        |  o--ro estimate-time?   fibrechannel:time-24hr-extended-type
        o--ro optical-loopback-test
        |  o--ro result?          result-type
        |  o--ro comments?        comments-type
        |  o--ro start-time?      fibrechannel:time-24hr-extended-type
        |  o--ro estimate-time?   fibrechannel:time-24hr-extended-type
        +--ro link-traffic-test
        |  +--ro result?          result-type
        |  +--ro comments?        comments-type
        |  +--ro start-time?      fibrechannel:time-24hr-extended-type
        |  +--ro estimate-time?   fibrechannel:time-24hr-extended-type
        +--ro start-time?                       string
        +--rw frame-count?                      yang:counter32
        +--rw frame-size?                       uint32
        +--rw time?                             string
        +--rw payload-pattern
        |  +--rw pattern?   string
        |  +--rw payload?   string
        +--rw fec
        |  +--rw enable?   enumeration
        |  +--ro active?   enumeration
        |  +--ro option?   enumeration
        +--ro rt-latency?                       uint32
        +--ro buffers-required?                 string
```

```
        +--ro end-time?                      string
        +--rw cr
        |  +--rw enable?   enumeration
        |  +--ro active?   enumeration
        |  +--ro option?   enumeration
        +--ro failure-report
        |  x--ro errors-detected-local?    enumeration
        |  x--ro errors-detected-remote?   enumeration
        +--ro failure-report-local-errors
        |  +--ro error*   error-stats-type
        +--ro failure-report-remote-errors
        |  +--ro error*   error-stats-type
        +--ro egress-power-loss
        |  +--ro tx?         decimal64
        |  +--ro rx?         decimal64
        |  +--ro loss?       decimal64
        |  +--ro comments?   string
        +--ro ingress-power-loss
           +--ro tx?         decimal64
           +--ro rx?         decimal64
           +--ro loss?       decimal64
           +--ro comments?   string
```

Examples

This section provides basic examples for using the brocade-fibrechannel-diagnostics module.

Retrieving the Test Status of a Diagnostic Port

This example uses a GET request to discover the status of the diagnostic port at the address “0/1”.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-diagnostics/fibrechannel-diagnostics/name/<port_ID>/state
```

URI

Notice that the slash character in the name resource is encoded as “%2f”.

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-diagnostics/fibrechannel-diagnostics/name/0%2f1/
state
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
  <fibrechannel-diagnostics>
    <name>0/1</name>
    <state>PASSED</state>
  </fibrechannel-diagnostics>
```

```
</Response>
```

Starting a Diagnostic Port Test

This example uses a PATCH request to start a diagnostic port test on port “0/1” with a frame count of 2 and a frame size of 2112 bytes.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-diagnostics/fibrechannel-diagnostics/name/<portID>/diagnostic-control/<instruction>
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-diagnostics/fibrechannel-diagnostics/name/0%2f1/diagnostic-control/1
```

Request Body

```
<frame-count>2</frame-count>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status in the headers.

Stopping a Diagnostic Port Test

This example uses a PATCH request to stop the diagnostic port test on port “0/1”.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-diagnostics/fibrechannel-diagnostics/name/<port_ID>/diagnostic-control/<instruction>
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-diagnostics/fibrechannel-diagnostics/name/0%2f1/diagnostic-control/0
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status in the headers.

brocade-fibrechannel-logical-switch

This module retrieves information of all logical switches in an operational chassis. This module also provides a logical switch configuration.

For a list of changes to this module by release, refer to the individual module YANG file.

It assumes a knowledge of Virtual Fabrics and logical switches as performed in Fabric OS. For more information, refer to *Managing Virtual Fabrics* in the *Brocade Fabric OS Administration Guide*.

NOTE
The brocade-fibrechannel-logical-switch module is supported in Fabric OS 8.2.0a and later.

Module Tree

This is the tree view of the module from the `brocade-fibrechannel-logical-switch.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Typesfor data type descriptions.

```
module: brocade-fibrechannel-logical-switch
  +--rw brocade-fibrechannel-logical-switch
     +--rw fibrechannel-logical-switch* [fabric-id] {fibrechannel:fibrechannel_virtual_fabric_platform}?
        +--rw fabric-id                 uint32
        +--ro switch-wwn?               fibrechannel:wwn-type
        x--rw base-switch-enabled?      uint8
        +--rw base-switch-enabled-v2?   boolean
        x--ro default-switch-status?    uint8
        +--ro default-switch?           boolean
        x--rw logical-isl-enabled?      uint8
        +--rw logical-isl-enabled-v2?   boolean
        x--rw ficon-mode-enabled?       uint8
        +--rw ficon-mode-enabled-v2?    boolean
        +--rw port-member-list
        |  +--rw port-member*   fibrechannel:slot-port-name-type
        +--rw ge-port-member-list
        |  +--rw port-member*   fibrechannel:slot-port-name-type
        +--rw port-index-members
           +--rw port-index*   fibrechannel:user-port-number-type
```

Examples

This section provides basic examples for using the brocade-fibrechannel-logical-switch module.

Retrieving Logical Switch Information

This example uses a GET request to retrieve the logical switch information.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-logical-switch>
        <fabric-id>128</fabric-id>
        <base-switch-enabled>0</base-switch-enabled>
        <switch-wwn>10:01:d1:1f:cc:1a:f1:1e</switch-wwn>
        <default-switch-status>1</default-switch-status>
        <default-switch>true</default-switch>
        <logical-isl-enabled>1</logical-isl-enabled>
        <ficon-mode-enabled>0</ficon-mode-enabled>
        <port-member-list>
            <port-member>0/4</port-member>
            <port-member>0/5</port-member>
            ...
            <port-member>0/62</port-member>
            <port-member>0/63</port-member>
        </port-member-list>
        <ge-port-member-list/>
        <port-index-members>
            <port-index>4</port-index>
            <port-index>5</port-index>
            ...
            <port-index>62</port-index>
            <port-index>63</port-index>
        </port-index-members>
    </fibrechannel-logical-switch>
    <fibrechannel-logical-switch>
        <fabric-id>79</fabric-id>
        <base-switch-enabled>0</base-switch-enabled>
        <switch-wwn>10:01:d1:1f:cc:1a:f1:1f</switch-wwn>
        <default-switch-status>0</default-switch-status>
        <default-switch>false</default-switch>
        <logical-isl-enabled>1</logical-isl-enabled>
        <ficon-mode-enabled>0</ficon-mode-enabled>
        <port-member-list>
            <port-member>0/20</port-member>
            <port-member>0/21</port-member>
            <port-member>0/22</port-member>
        </port-member-list>
        <ge-port-member-list/>
        <port-index-members>
            <port-index>20</port-index>
            <port-index>21</port-index>
            <port-index>22</port-index>
        </port-index-members>
    </fibrechannel-logical-switch>
</Response>
```

Retrieving Information for a Logical Switch Instance

This example uses a GET request to retrieve logical switch information for fabric ID 10.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch/fabric-id/fabric_id
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch/fabric-id/10
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-logical-switch>
        <fabric-id>79</fabric-id>
        <base-switch-enabled>0</base-switch-enabled>
        <switch-wwn>10:01:d1:1f:cc:1a:f1:1f</switch-wwn>
        <default-switch-status>0</default-switch-status>
        <default-switch>false</default-switch>
        <logical-isl-enabled>1</logical-isl-enabled>
        <ficon-mode-enabled>0</ficon-mode-enabled>
        <port-member-list>
            <port-member>0/20</port-member>
            <port-member>0/21</port-member>
            <port-member>0/22</port-member>
        </port-member-list>
        <ge-port-member-list/>
        <port-index-members>
            <port-index>20</port-index>
            <port-index>21</port-index>
            <port-index>22</port-index>
        </port-index-members>
    </fibrechannel-logical-switch>
</Response>
```

Creating a Logical Switch

This example uses a POST request to create logical switch 20.

> **NOTE**
> REST requests time out after 3 minutes; therefore, it is recommended that you do not create more than three logical switches in one POST request.

Structure

```
POST <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
POST https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

```
<fibrechannel-logical-switch>
   <fabric-id>20</fabric-id>
</fibrechannel-logical-switch>
```

Response Body

When the operation is successful, there is no response, and a “201 Created” status appears in the headers.

Assigning Ports to a Logical Switch

You can use a POST or PATCH request to assign ports to a logical switch. You must use POST to assign ports to a new logical switch. You can use both a POST request or a PATCH request to assign ports to an existing logical switch. However, when you use a PATCH request to assign ports to an existing logical switch, the PATCH request replaces the existing port list with the new port list.

Assigning Ports to a Logical Switch

NOTE
You cannot use a PATCH request to assign ports to the default logical switch. You cannot use a PATCH or POST request to send an empty port list to a logical switch.

This example uses a POST request to assign three ports to a logical switch 20.

Structure

```
POST <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
POST https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

```
<fibrechannel-logical-switch>
   <fabric-id>20</fabric-id>
      <port-member-list>
         <port-member>0/1</port-member>
         <port-member>0/2</port-member>
         <port-member>0/3</port-member>
      </port-member-list>
</fibrechannel-logical-switch>
```

Response Body

When the operation is successful, there is no response, and a “201 Created” status appears in the headers.

Creating a FICON Logical Switch

This example uses a POST request to create FICON logical switch 20, enable FICON mode, and assign ports to the new FICON logical switch.

> **NOTE**
> REST requests time out after 3 minutes; therefore, it is recommended that you do not create more than three logical switches in one POST request.

Structure

```
POST <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
POST https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

```
<fibrechannel-logical-switch>
    <fabric-id>20</fabric-id>
          <ficon-mode-enabled>1</ficon-mode-enabled>
          <port-member-list>
                <port-member>0/1</port-member>
                <port-member>0/2</port-member>
                <port-member>0/3</port-member>
          </port-member-list>
</fibrechannel-logical-switch>
```

Response Body

When the operation is successful, there is no response, and a “201 Created” status appears in the headers.

To complete the FICON configuration and configure FICON CUP, refer to the *Brocade Fabric OS FICON Configuration Guide*. The brocade-ficon module allows management of some FICON Control Unit Port (CUP) and FICON logical switch parameters as well as providing FICON-related information from the Management Server (such as RNID, LIRR, and RLIR data). To use the brocade-ficon module, refer to brocade-ficon.

Deleting a Logical Switch

This example uses a DELETE request to delete a logical switch 23.

> **NOTE**
> You cannot delete the default logical switch.

Structure

```
DELETE <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

```
<fibrechannel-logical-switch>
            <fabric-id>23</fabric-id>
```

```
</fibrechannel-logical-switch>
```

Response Body

When the operation is successful, there is no response, and a “204 No Content” status appears in the headers.

Removing Ports from a Logical Switch

This example uses a DELETE request to remove two ports from logical switch 20.

Structure

```
DELETE <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

```
<fibrechannel-logical-switch>
   <fabric-id>20</fabric-id>
   <port-member-list>
      <port-member>0/1</port-member>
      <port-member>0/2</port-member>
   </port-member-list>
</fibrechannel-logical-switch>
```

Response Body

When the operation is successful, there is no response, and a “204 No Content” status appears in the headers.

Enabling the Base Switch

You must use a POST request to enable a new logical switch as the base switch. You must use a PATCH request to enable an existing logical switch as the base switch.

> **NOTE**
> Only one logical switch can be configured as the base switch per chassis.

This example uses a POST request to enable a new logical switch 23 as the base switch.

Structure

```
POST <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
POST https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

```
<fibrechannel-logical-switch>
   <fabric-id>23</fabric-id>
   <base-switch-enabled>1</base-switch-enabled>
</fibrechannel-logical-switch>
```

Response Body

When the operation is successful, there is no response, and a “201 Created” status appears in the headers.

Disabling the Base Switch

You must use a PATCH request to disable an existing logical switch as the base switch.

This example uses a PATCH request to disable logical switch 23 as the base switch.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

```
<fibrechannel-logical-switch>
   <fabric-id>23</fabric-id>
   <base-switch-enabled>0</base-switch-enabled>
</fibrechannel-logical-switch>
```

Response Body

When the operation is successful, there is no response, and a “204 No Content” status appears in the headers.

Enabling FICON Mode on a Logical Switch

You must use a POST request to enable FICON mode on a new logical switch. You must use a PATCH request to enable FICON mode on an existing logical switch.

This example uses a POST request to create new logical switch 23 with FICON mode enabled.

Structure

```
POST <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
POST https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

```
<fibrechannel-logical-switch>
   <fabric-id>23</fabric-id>
   <ficon-mode-enabled>1</ficon-mode-enabled>
</fibrechannel-logical-switch>
```

Response Body

When the operation is successful, there is no response, and a “201 Created” status appears in the headers.

Enabling a Logical ISL on a Logical Switch

Logical ISLs can be disabled during logical switch creation with the POST request and can later be enabled with a PATCH request.

This example uses a PATCH request to enable logical ISLs on logical switch 23.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

```
<fibrechannel-logical-switch>
      <fabric-id>23</fabric-id>
      <logical-isl-enabled>1</logical-isl-enabled>
</fibrechannel-logical-switch>
```

Response Body

When the operation is successful, there is no response, and a “204 No Content” status appears in the headers.

Disabling a Logical ISL on a Logical Switch

You can disable a logical ISL only during logical switch creation. You must use a POST request to disable a logical ISL and create a new logical switch.

NOTE
You cannot disable a logical ISL on an existing logical switch.

This example uses a POST request to disable a logical ISL on new logical switch 24.

Structure

```
POST <base_URI>/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

URI

```
POST https://10.10.10.10/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
```

Request Body

```
<fibrechannel-logical-switch>
   <fabric-id>24</fabric-id>
   <logical-isl-enabled>0</logical-isl-enabled>
      <port-member-list>
         <port-member>0/1</port-member>
         <port-member>0/2</port-member>
         <port-member>0/3</port-member>
      </port-member-list>
</fibrechannel-logical-switch>
```

Response Body

When the operation is successful, there is no response, and a “201 Created” status appears in the headers.

brocade-fibrechannel-routing

The Fibre Channel Routing module enables you to configure FC routing between two or more fabrics without merging those fabrics.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-fibrechannel-routing.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-fibrechannel-routing
  +--rw brocade-fibrechannel-routing
     +--rw routing-configuration {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw maximum-lsan-count?                    uint32
     |  +--rw backbone-fabric-id?                    fibrechannel:fabric-id-type
     |  +--rw shortest-ifl?                          boolean
     |  +--rw lsan-enforce-tags
     |  |  +--rw tag*   string
     |  +--rw lsan-speed-tag?                        string
     |  +--rw migration-mode?                        boolean
     |  +--rw persistent-translate-domain-enabled?   boolean
     |  +--rw backbone-fabric-alias?                 string
     +--ro lsan-zone* []
     |  +--ro edge-fabric-id?   fibrechannel:fabric-id-type
     |  +--ro zone-name?        zone:zoning-name-type
     |  +--ro members
     |     +--ro member*   fibrechannel:wwn-type
     +--ro lsan-device* []
     |  +--ro source-fabric-id?     fibrechannel:fabric-id-type
     |  +--ro device-wwn?           fibrechannel:wwn-type
     |  +--ro proxy-pid?            fibrechannel:fcid-hex-string-type
     |  +--ro imported-fabric-id?   fibrechannel:fabric-id-type
     |  +--ro physical-pid?         fibrechannel:fcid-hex-string-type
     |  +--ro device-status?        fcr-device-state
     +--rw edge-fabric-alias* [edge-fabric-id] {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw edge-fabric-id    fibrechannel:fabric-id-type
     |  +--rw alias-name?       string
     +--ro fibrechannel-router* []
     |  +--ro domain-id?                  fibrechannel:domain-id-type
     |  +--ro router-wwn?                 fibrechannel:wwn-type
     |  +--ro ipv4-address?               union
     |  +--ro ipv6-address?               union
     |  +--ro switch-user-friendly-name?   string
     +--ro router-statistics {fibrechannel:fibrechannel_router_platform}?
     |  +--ro lsan-zones-in-use?            uint32
     |  +--ro maximum-lsan-devices?         uint32
     |  +--ro lsan-devices-in-use?          uint32
     |  +--ro maximum-proxy-device-slots?   uint32
     |  +--ro proxy-device-slots-in-use?    uint32
     |  +--ro maximum-proxy-devices?        uint32
     |  +--ro maximum-nr-ports?             uint32
```

```
   +--rw proxy-config* [imported-fabric-id device-wwn]
   |  +--rw imported-fabric-id    fibrechannel:fabric-id-type
   |  +--rw device-wwn            fibrechannel:wwn-type
   |  +--rw proxy-device-slot?    fibrechannel:area-address-type
   +--rw translate-domain-config* [imported-fabric-id exported-fabric-id]
   |  +--rw imported-fabric-id               fibrechannel:fabric-id-type
   |  +--rw exported-fabric-id               fibrechannel:fabric-id-type
   |  +--rw preferred-translate-domain-id?   fibrechannel:domain-id-type
   |  +--ro owner-front-domain-id?           fibrechannel:domain-id-type
   |  +--ro translate-wwn?                   fibrechannel:wwn-type
   |  +--ro front-domain-owner?              fibre-channel-router-owner
   +--rw stale-translate-domain* [imported-fabric-id stale-translate-domain-id]
      +--rw imported-fabric-id           fibrechannel:fabric-id-type
      +--rw stale-translate-domain-id    fibrechannel:domain-id-type
      +--ro owner-front-domain-id?       fibrechannel:domain-id-type
      +--ro front-domain-owner?          fibre-channel-router-owner
      +--ro dual-backbone?               boolean
```

Examples

This section provides basic examples for using the brocade-fibrechannel-routing module.

Configuring FC-FC Routing

NOTE
Before you set or modify the backbone-fabric-id, the switch and FCR must be disabled.

This example uses a PATCH request to configure the following options for FC-FC routing.

- Set the maximum LSAN count to 7500.
- Set the backbone fabric ID to 100.
- Enable the shortest IFL mode.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-routing/routing-configuration
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-routing/routing-configuration
```

Request Body

```
<routing-configuration>
    <maximum-lsan-count>7500</maximum-lsan-count>
    <backbone-fabric-id>100</backbone-fabric-id>
    <shortest-ifl>true</shortest-ifl>
</routing-configuration>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

Setting the Maximum LSAN Count

This example uses a PATCH request to set the maximum LSAN count to 3000.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-routing/routing-configuration
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-routing/routing-configuration
```

Request Body

```
 <routing-configuration>
    <maximum-lsan-count>3000</maximum-lsan-count>
</routing-configuration>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Assigning a Backbone Fabric ID

This example uses a PATCH request to assign a backbone fabric ID to 101.

NOTE
You must disable the switch to change the backbone fabric ID.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-routing/routing-configuration
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-routing/routing-configuration
```

Request Body

```
<routing-configuration>
    <backbone-fabric-id>101</backbone-fabric-id>
</routing-configuration>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Enabling Shortest IFL Mode

You can direct traffic flow to take the shortest path between host and target when multiple FC routers in the backbone are connected to an edge fabric.

This example uses a PATCH request to enable the shortest IFL mode.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-routing/routing-configuration
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-routing/routing-configuration
```

Request Body

```
   <routing-configuration>
        <shortest-ifl>true</shortest-ifl>
    </routing-configuration>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Disabling Shortest IFL Mode

This example uses a PATCH request to disable the shortest IFL mode.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-routing/routing-configuration
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-routing/routing-configuration
```

Request Body

```
 <routing-configuration>
    <shortest-ifl>true</shortest-ifl>
 </routing-configuration>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Configuring Enforce LSAN Tags

Use the Enforce tag to achieve better scalability in the FC router by limiting the number of LSAN zones that are enforced in that FC router.

This example uses a POST request to configure a enforce LSAN tags for lsan_abc, lsan_xyz, and lsan_fab1.

NOTE
You must disable the switch to change the LSAN enforce tags.

Structure

```
POST <base_URI>/running/brocade-fibrechannel-routing/routing-configuration
```

URI

```
POST https://10.10.10.10/rest/running/brocade-fibrechannel-routing/routing-configuration
```

Request Body

```
<routing-configuration>
     <lsan-enforce-tags>
         <lsan_abc>
         <lsan_xyz>
         <lsan_fab1>
     </lsan-enforce-tags>
</routing-configuration>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

Configuring a Speed LSAN Tag

Use the Speed tag to speed up the discovery process by importing devices into the remote edge fabrics when the devices come online, regardless of the state of the host.

This example uses a PATCH request to set the maximum LSAN count.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-routing/routing-configuration
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-routing/routing-configuration
```

Request Body

```
<routing-configuration>
        <lsan-speed-tag>mcdt</lsan-speed-tag>
</routing-configuration>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

Deleting an LSAN Tag

This example uses a DELETE request to delete the test LSAN tag.

NOTE
You must disable the switch to remove the LSAN enforce tags.

Structure

```
DELETE <base_URI>/running/brocade-fibrechannel-routing/routing-configuration/lsan-enforce-tag/tags/<tag_name>
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-fibrechannel-routing/routing-configuration/lsan-enforce-tags/
tags/test
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Viewing the LSAN Zone Information

This example uses a GET request to retrieve LSAN zone data for the device to which the request is sent.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-routing/lsan-zone
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-routing/lsan-zone
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <lsan-zone>
        <edge-fabric-id>100</edge-fabric-id>
        <zone-name>lsan_68162_Violin</zone-name>
        <members>
            <member>10:00:1c:2c:ff:14:ec:00</member>
            <member>10:00:1c:2c:ff:14:ec:01</member>
            <member>10:00:1c:2c:ff:14:d8:03</member>
            <member>21:00:00:1b:17:18:da:f4</member>
        </members>
    </lsan-zone>
            ...
    <lsan-zone>
        <edge-fabric-id>128</edge-fabric-id>
        <zone-name>lsan_68163_Violin</zone-name>
        <members>
            <member>10:00:00:00:c9:f8:04:05</member>
            <member>10:00:00:00:c9:f8:04:06</member>
            <member>21:00:00:1b:10:11:d8:07</member>
            <member>21:00:00:1b:10:11:da:f8</member>
        </members>
    </lsan-zone>
    <lsan-zone>
        <edge-fabric-id>128</edge-fabric-id>
```

```
        <zone-name>lsan_68164_Violin</zone-name>
        <members>
            <member>21:00:00:0e:1e:07:03:09</member>
            <member>21:00:00:0e:1e:07:04:10</member>
            <member>21:00:00:1b:10:11:d8:11</member>
            <member>21:00:00:1b:10:11:da:f2</member>
        </members>
    </lsan-zone>
</Response>
```

Viewing the LSAN Device Information

This example uses a GET request to retrieve LSAN data for the device to which the request is sent.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-routing/lsan-device
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-routing/lsan-device
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <lsan-device>
        <source-fabric-id>100</source-fabric-id>
        <device-wwn>10:00:00:0e:1e:17:04:01</device-wwn>
        <physical-pid>0xec0500</physical-pid>
        <device-status>exist</device-status>
    </lsan-device>
    <lsan-device>
        <source-fabric-id>100</source-fabric-id>
        <device-wwn>21:00:00:0e:1e:17:03:02</device-wwn>
        <physical-pid>0x2baa00</physical-pid>
        <device-status>exist</device-status>
    </lsan-device>
...
    <lsan-device>
        <source-fabric-id>100</source-fabric-id>
        <device-wwn>21:00:00:1b:10:11:da:f3</device-wwn>
        <physical-pid>0xeb2901</physical-pid>
        <device-status>exist</device-status>
    </lsan-device>
...
    <lsan-device>
        <source-fabric-id>100</source-fabric-id>
        <device-wwn>20:af:00:a0:10:9f:76:a4</device-wwn>
```

```
        <physical-pid>0xed0203</physical-pid>
        <proxy-pid>0xcbf001</proxy-pid>
        <imported-fabric-id>128</imported-fabric-id>
        <device-status>imported</device-status>
    </lsan-device>
...
    <lsan-device>
        <source-fabric-id>100</source-fabric-id>
        <device-wwn>21:00:00:1b:10:11:d8:05</device-wwn>
        <physical-pid>0xeb0001</physical-pid>
        <proxy-pid>0xcbf401</proxy-pid>
        <imported-fabric-id>128</imported-fabric-id>
        <device-status>imported</device-status>
    </lsan-device>
</Response>
```

Configuring an Alias for an Edge Fabric

This example uses a PATCH request to configure test11 as an alias name for the edge fabric ID 100.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-routing/edge-fabric-alias
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-routing/edge-fabric-alias
```

Request Body

```
<edge-fabric-alias>
   <edge-fabric-id>100</edge-fabric-id>
   <alias-name>test11</alias-name>
</edge-fabric-alias>
```

Response Body

When the operation is successful, the response has an empty response body, and a “204 No Content” status appears in the header.

Enabling FC Routing

This example uses a PATCH request to enable FC routing on the device to which the request is sent.

Structure

```
PATCH <base_URI>/running/brocade-chassis/chassis/fcr-enabled/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-chassis/chassis/fcr-enabled/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Disabling FC Routing

This example uses a PATCH request to disable FC routing on the device to which the request is sent.

Structure

```
PATCH <base_URI>/running/brocade-chassis/chassis/chassis/fcr-enabled/false
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-chassis/chassis/chassis/fcr-enabled/false
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Configuring the Fabric ID for the Edge Fabric

This example uses a PATCH request to to configure the edge fabric ID to 49. Note that the EX_Port must be enabled before you can configure the Fabric ID for the edge fabric.

Structure

```
PATCH <base_URI>/running/brocade-interface/fibrechannel/edge-fabric-id/fabric_id
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-interface/fibrechannel/49
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Enabling a Port as an EX_Port

This example uses a PATCH request to to enable the EX_Port for the edge fabric. Note that Fibre Channel switch Native mode must be supported.

Structure

```
PATCH <base_URI>/running/brocade-interface/fibrechannel/ex-port-enabled/1
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-interface/ex-port-enabled/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Configuring the Preferred Domain ID

This example uses a PATCH request to to configure the preferred domain ID to 160. Note that the EX_Port must be enabled before you can configure the Fabric ID for the edge fabric.

Structure

```
PATCH <base_URI>/running/brocade-interface/fibrechannel/preferred-front-domain-id/preferred_domain_id
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-interface/fibrechannel/preferred-front-domain-id/160
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fibrechannel-switch

This module provides a detailed view of the switch being queried. If virtual fabrics are enabled, the request can include a query parameter (VFID) for the desired virtual fabric. If no query parameter is specified and virtual fabrics are enabled, the default virtual fabric values are returned.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-fibrechannel-switch.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Typesfor data type descriptions.

```
module: brocade-fibrechannel-switch
  +--rw brocade-fibrechannel-switch
     +--rw fibrechannel-switch* [name]
     |  +--rw name                                  fibrechannel:wwn-type
     |  +--rw domain-id?                            fibrechannel:domain-id-type
  {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw user-friendly-name?                   string
     |  x--ro fcid?                                 fibrechannel:fcid-type
  {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro fcid-hex?                             fibrechannel:fcid-hex-string-type
  {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro vf-id?                               int16 {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro principal?                            uint8 {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro is-principal?                         boolean {fibrechannel:fibrechannel_switch_platform}?
     |  x--rw enabled-state?                        uint32
     |  +--rw is-enabled-state?                     boolean
     |  +--ro operational-status?                   uint32
     |  +--ro operational-status-string?            operational-status-type
     |  x--ro up-time?                              yang:timeticks
     |  +--ro model?                                string
     |  +--ro firmware-version?                     string
     |  x--rw ip-address
     |  |  x--rw ip-address*   union
     |  x--rw ip-static-gateway-list
     |  |  x--rw ip-static-gateway*   fru:ip-gateway-type
     |  x--rw subnet-mask?                          inet:ipv4-address
     |  +--rw ipfc-address?                         union {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw ipfc-subnet-mask?                     union {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw domain-name?                          string
     |  +--rw dns-servers
     |  |  +--rw dns-server*   union
     |  +--rw fabric-user-friendly-name?            string {fibrechannel:fibrechannel_switch_platform}?
     |  x--rw ag-mode?                              uint8
     |  +--rw ag-mode-string?                       fibrechannel:ag-mode-type
     |  +--rw banner?                               string
     |  +--rw in-order-delivery-enabled?            boolean {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw dynamic-load-sharing?                 fibrechannel:dynamic-load-sharing-level-type
  {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw advanced-performance-tuning-policy?   fibrechannel:advanced-performance-tuning-policy-type
  {fibrechannel:fibrechannel_switch_platform}?
```

```
     |  +--rw lacp-system-priority?                uint32
     |  +--ro lacp-system-mac-address?             ietf:mac-address
     |  +--rw switch-persistent-enabled?           boolean
     |  +--ro switch-addressing-mode?              switch-addressing-mode-type
 {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro switch-beacon-enabled?               boolean
     |  +--ro high-integrity-fabric-mode-enabled?  boolean {fibrechannel:fibrechannel_switch_platform}?
     +--ro topology-domain* [domain-id] {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro domain-id                   fibrechannel:domain-id-type
     |  +--ro path-count?                 yang:gauge32
     |  +--ro metric?                     uint32
     |  +--ro is-local-translate-domain?  boolean
     |  +--ro out-ports
     |     +--ro port*   fibrechannel:user-port-number-type
     +--ro topology-route* [] {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro domain-id?   fibrechannel:domain-id-type
     |  +--ro hop-count?   uint32
     |  +--ro out-port?    fibrechannel:user-port-number-type
     |  +--ro in-ports
     |     +--ro port*   fibrechannel:user-port-number-type
     +--ro topology-error* [] {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro domain-id?    fibrechannel:domain-id-type
     |  +--ro error-type?   fibrechannel:topology-report-error-type
     +--ro switch-fabric-statistics* [] {fibrechannel:fibrechannel_switch_platform}?
        +--ro fabric-incident?       fabric-incident-type
        +--ro segmentation-reason?   segmentation-reason-type
        +--ro last-event?            boolean
        +--ro incident-count?        yang:zero-based-counter32
        +--ro latest-port-index?     fibrechannel:port-number-type
        +--ro time-stamp?            yang:date-and-time
```

Examples

This section provides basic examples for using the brocade-fibrechannel-switch module.

Retrieving Switch Resource Information

This example uses a GET request to retrieve the switch resources.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-switch/fibrechannel-switch/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-switch/fibrechannel-switch/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
```

```
    "Response": {
        "fibrechannel-switch": [
            {
                "name": "10:00:c4:f5:7c:16:9f:78",
                "domain-id": 71,
                "fcid": 16776263,
                "fcid-hex": "0xfffc47",
                "user-friendly-name": "wedge_71",
                "enabled-state": 2,
                "is-enabled-state": true,
                "operational-status": 2,
                "banner": "",
                "up-time": 24580,
                "domain-name": "testv6.bsnlab.broadcom.net.",
                "dns-servers": {
                    "dns-server": [
                        "2620:100:4:e200::180"
                    ]
                },
                "principal": 0,
                "ip-address": {
                    "ip-address": [
                        "10.38.18.71",
                        "2620:100:4:f801::f1e/64"
                    ]
                },
                "subnet-mask": "255.255.240.0",
                "ipfc-address": "1.2.1.2",
                "ipfc-subnet-mask": "255.255.248.0",
                "model": "162.0",
                "firmware-version": "v9.1.0IPFC_REST",
                "ip-static-gateway-list": {
                    "ip-static-gateway": [
                        "10.38.16.1"
                    ]
                },
                "vf-id": 128,
                "fabric-user-friendly-name": "new",
                "ag-mode": 1,
                "dynamic-load-sharing": "lossless-dls",
                "in-order-delivery-enabled": false,
                "advanced-performance-tuning-policy": "exchange-based"
            }
        ]
    }
}
```


Retrieving Switch Resource Information from a Virtual Fabric

This example uses a GET request to retrieve the switch resource information for a switch in the virtual fabric with a VFID of 20.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-switch/fibrechannel-switch?vf-id=<vf-id#>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-switch/fibrechannel-switch?vf-id=20
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-switch>
        <name>10:00:c1:f1:1c:1a:ac:6c</name>
        <domain-id>43</domain-id>
        <fcid>16776235</fcid>
        <fcid-hex>0xfffc2b</fcid-hex>
        <user-friendly-name>66-243_X6-4_Venator</user-friendly-name>
        <enabled-state>2</enabled-state>
        <is-enabled-state>true</is-enabled-state>
        <operational-status>2</operational-status>
        <banner/>
        <up-time>12766</up-time>
        <domain-name>brm.bsnlab.broadcom.net</domain-name>
        <dns-servers>
            <dns-server>192.10.10.10</dns-server>
            <dns-server>192.10.10.20</dns-server>
        </dns-servers>
        <principal>0</principal>
        <ip-address>
            <ip-address>10.38.66.243</ip-address>
        </ip-address>
        <subnet-mask>255.255.240.0</subnet-mask>
        <model>165.0</model>
        <firmware-version>v9.0.0</firmware-version>
        <ip-static-gateway-list>
            <ip-static-gateway>10.10.11.12</ip-static-gateway>
        </ip-static-gateway-list>
        <vf-id>20</vf-id>
        <fabric-user-friendly-name>East</fabric-user-friendly-name>
        <ag-mode>0</ag-mode>
        <dynamic-load-sharing>lossless-dls</dynamic-load-sharing>
        <in-order-delivery-enabled>false</in-order-delivery-enabled>
        <advanced-performance-tuning-policy>exchange-based</advanced-performance-tuning-policy>
        <lacp-system-mac-address>c4:f5:7c:59:ac:6e</lacp-system-mac-address>
        <lacp-system-priority>32768</lacp-system-priority>
    </fibrechannel-switch>
</Response>
```

Setting the User-Friendly Name for a Switch

This example uses a PATCH request to set the user-friendly name for the switch “10:10:10:10:1c:16:98:14” to “My_Switch_1”.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-switch/fibrechannel-switch/name/switch_name/user-friendly-name/friendly_name
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-switch/fibrechannel-switch/name/10:10:10:10:1c:16:98:14/user-friendly-name/My_Switch_1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No content” status in the header.

Setting the Login Banner for a Switch

This example uses a PATCH request to set the login banner for the switch “10:10:10:10:1c:16:98:14” to “Login Banner-REST”.

Structure

```
PATCH <base_URI>/running/brocade-fibrechannel-switch/fibrechannel-switch
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fibrechannel-switch/fibrechannel-switch
```

Request Body

```
<fibrechannel-switch>
    <name>10:10:10:10:7c:c0:ad:72</name>
    <banner>Login Banner-REST</banner>
</fibrechannel-switch>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

brocade-fibrechannel-trunk

This module provides a detailed view of all trunks in the switch in native mode as well as the members of the individual trunks. It can also provide traffic performance and bandwidth information. For F_Port static trunks, you can configure a trunk index that is persistent and has the same value across all members of the trunk. Even if the master port changes, the trunk area is derived from the target index.

For a list of changes to this module by release, refer to the individual module YANG file.

It assumes a knowledge of trunking as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Module Tree

This is the tree view of the module from the `brocade-fibrechannel-trunk.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Typesfor data type descriptions.

```
module: brocade-fibrechannel-trunk
  +--rw brocade-fibrechannel-trunk
     +--ro trunk* [group source-port] {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro group                   fibrechannel:user-port-number-type
     |  +--ro source-port             fibrechannel:user-port-number-type
     |  +--ro master?                 boolean
     |  +--ro destination-port?       fibrechannel:user-port-number-type
     |  +--ro neighbor-wwn?           fibrechannel:wwn-type
     |  +--ro neighbor-switch-name?   string
     |  +--ro neighbor-domain-id?     fibrechannel:domain-id-type
     |  +--ro deskew?                 fibrechannel:deskew-type
     |  +--ro trunk-type?             fibrechannel:fibrechannel-trunk-type
     +--ro performance* [group] {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro group                fibrechannel:user-port-number-type
     |  +--ro tx-bandwidth?        uint32
     |  +--ro tx-capacity?         decimal64
     |  +--ro tx-utilization?      uint64
     |  x--ro tx-throughput?       uint64
     |  +--ro tx-percentage?       fibrechannel:percentage-type
     |  +--ro rx-bandwidth?        uint32
     |  +--ro rx-capacity?         decimal64
     |  +--ro rx-utilization?      uint64
     |  x--ro rx-throughput?       uint64
     |  +--ro rx-percentage?       fibrechannel:percentage-type
     |  +--ro txrx-bandwidth?      uint32
     |  +--ro txrx-capacity?       decimal64
     |  +--ro txrx-utilization?    uint64
     |  x--ro txrx-throughput?     uint64
     |  +--ro txrx-percentage?     fibrechannel:percentage-type
     +--rw trunk-area* [trunk-index] {fibrechannel:fibrechannel_switch_platform}?
        +--rw trunk-index      fibrechannel:user-port-number-type
        +--rw trunk-members
        |  +--rw trunk-member*   fibrechannel:slot-port-name-type
        +--ro master-port?     fibrechannel:slot-port-name-type
        +--ro trunk-active?    boolean
```

Examples

This section provides basic examples for using the brocade-fibrechannel-trunk module.

Viewing Trunk Settings for a Port

The following example uses the GET request to view the trunk settings for port 0/88.

Structure

```
GET <base_URI>/running/brocade-interface/fibrechannel/name/fibre_channel_interface_name/trunk-port-enabled/
```

URI

Notice that the slash character in the name resource is encoded as “%2f”.

```
GET https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/0%2f88/trunk-port-enabled/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel>
        <name>0/88</name>
        <trunk-port-enabled>1</trunk-portenabled>
    </fibrechannel>
</Response>
```

Enabling Trunking on a Port

The following example uses the PATCH request to enable trunking for port 0/88.

Structure

```
 PATCH <base_URI>/running/brocade-interface/fibrechannel/name/fibre_channel_interface_name/trunk-port-
enabled/1
```

URI

Notice that the slash character in the name resource is encoded as “%2f”.

```
PATCH https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/0%2f88/trunk-port-enabled/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content" status appears in the header.

Viewing Trunk Data for a Source Port

The following example uses the GET request to view trunk data for source port 88 in trunk group 1.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-trunk/trunk/group/group_number/source-port/port_number
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-trunk/trunk/group/1/source-port/88
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <trunk>
        <group>1</group>
        <source-port>88</source-port>
        <destination-port>24</destination-port>
        <neighbor-wwn>10:00:c4:f5:7c:00:c5:30</neighbor-wwn>
        <neighbor-switch-name>G630_066_232</neighbor-switch-name>
        <neighbor-domain-id>232</neighbor-domain-id>
        <deskew>15</deskew>
        <master>true</master>
        <trunk-type>inter-switch-link</trunk-type>
    </trunk>
</Response>
```

Viewing Trunk Performance Data

The following example uses the GET request to view performance data for all trunks on the switch.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-trunk/performance
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-trunk/performance
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <performance>
```

```
        <group>1</group>
        <tx-bandwidth>53</tx-bandwidth>
        <tx-capacity>50.00</tx-capacity>
        <tx-throughput>1982724144</tx-throughput>
        <tx-utilization>1982724144</tx-utilization>
        <tx-percentage>3.97</tx-percentage>
        <rx-bandwidth>53</rx-bandwidth>
        <rx-capacity>50.00</rx-capacity>
        <rx-throughput>1849092112</rx-throughput>
        <rx-utilization>1849092112</rx-utilization>
        <rx-percentage>3.70</rx-percentage>
        <txrx-bandwidth>106</txrx-bandwidth>
        <txrx-capacity>100.00</txrx-capacity>
        <txrx-throughput>3831816256</txrx-throughput>
        <txrx-utilization>3831816256</txrx-utilization>
        <txrx-percentage>3.83</txrx-percentage>
    </performance>
...
    <performance>
        <group>3</group>
        <tx-bandwidth>53</tx-bandwidth>
        <tx-capacity>50.00</tx-capacity>
        <tx-throughput>1607105424</tx-throughput>
        <tx-utilization>1607105424</tx-utilization>
        <tx-percentage>3.21</tx-percentage>
        <rx-bandwidth>53</rx-bandwidth>
        <rx-capacity>50.00</rx-capacity>
        <rx-throughput>1666583248</rx-throughput>
        <rx-utilization>1666583248</rx-utilization>
        <rx-percentage>3.33</rx-percentage>
        <txrx-bandwidth>106</txrx-bandwidth>
        <txrx-capacity>100.00</txrx-capacity>
        <txrx-throughput>3273688672</txrx-throughput>
        <txrx-utilization>3273688672</txrx-utilization>
        <txrx-percentage>3.27</txrx-percentage>
    </performance>
</Response>
```

Viewing Performance Data for a Specific Trunk Group

The following example uses the GET request to view performance data for trunk group 1.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-trunk/performance/group/group_number
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-trunk/performance/group/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <performance>
        <group>1</group>
        <tx-bandwidth>53</tx-bandwidth>
        <tx-capacity>50.00</tx-capacity>
        <tx-throughput>2653980384</tx-throughput>
        <tx-utilization>2653980384</tx-utilization>
        <tx-percentage>5.31</tx-percentage>
        <rx-bandwidth>53</rx-bandwidth>
        <rx-capacity>50.00</rx-capacity>
        <rx-throughput>699220720</rx-throughput>
        <rx-utilization>699220720</rx-utilization>
        <rx-percentage>1.40</rx-percentage>
        <txrx-bandwidth>106</txrx-bandwidth>
        <txrx-capacity>100.00</txrx-capacity>
        <txrx-throughput>3353201104</txrx-throughput>
        <txrx-utilization>3353201104</txrx-utilization>
        <txrx-percentage>3.35</txrx-percentage>
    </performance>
</Response>
```

Viewing Data for all Trunk Area Groups

The following example uses the GET request to obtain data for all port trunk area groups on a switch. The data includes both active and in-active configured trunk area groups.

Structure

```
GET <base_URI>/running/brocade-fibrechannel-trunk/trunk-area/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-trunk/trunk-area/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <trunk-area>
        <trunk-index>0</trunk-index>
        <trunk-active>true</trunk-active>
        <master-port>0/0</master-port>
        <trunk-members>
            <trunk-member>0/0</trunk-member>
            <trunk-member>0/6</trunk-member>
```

```
            <trunk-member>0/7</trunk-member>
        </trunk-members>
    </trunk-area>
    <trunk-area>
        <trunk-index>40</trunk-index>
        <trunk-active>false</trunk-active>
        <master-port/>
        <trunk-members>
            <trunk-member>0/40</trunk-member>
            <trunk-member>0/41</trunk-member>
        </trunk-members>
    </trunk-area>
</Response>
```

Configuring a Trunk Area Group

The following example uses the POST request to configure a port trunk area group 0 using ports 0/0, 0/6, and 0/7. Note that you must specify the trunk members with the POST request or the operation fails.

Structure

```
POST <base_URI>/running/brocade-fibrechannel-trunk/trunk-area/trunk-index/trunk-index-number
```

URI

```
POST https://10.10.10.10/rest/running/brocade-fibrechannel-trunk/trunk-area/trunk-index/0
```

Request Body

```
<trunk-members>
   <trunk-member>0/0</trunk-member>
   <trunk-member>0/6</trunk-member>
   <trunk-member>0/7</trunk-member>
</trunk-members>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created" status appears in the header.

Deleting a Trunk Area Group

The following example uses the DELETE request to delete port trunk area group 1 which contains ports 0/1, 0/2, and 0/3.

To delete an entire port trunk area, all members must be included in the Request Body.

Structure

```
DELETE <base_URI>/running/brocade-fibrechannel-trunk/trunk-area/trunk-index/trunk_index_number
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-fibrechannel-trunk/trunk-area/trunk-index/1
```

Request Body

```
<trunk-members>
```

```
    <trunk-member>0/1</trunk-member>
    <trunk-member>0/2</trunk-member>
    <trunk-member>0/3</trunk-member>
</trunk-members>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content" status appears in the header.

Deleting Members From an Existing Trunk Area Group

The following example uses the DELETE request to remove trunk members 0/6 and 0/7 from trunk area group 1 which contains ports 0/1, 0/6, and 0/7. Note that you must specify the trunk members with the DELETE request or the operation fails.

Structure

```
DELETE <base_URI>/running/brocade-fibrechannel-trunk/trunk-area/trunk-index/trunk_index_number
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-fibrechannel-trunk/trunk-area/trunk-index/1
```

Request Body

```
<trunk-members>
   <trunk-member>0/6</trunk-member>
   <trunk-member>0/7</trunk-member>
</trunk-members>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content" status appears in the header.

brocade-ficon

This module allows the management of some FICON Control Unit Port (CUP) parameters as well as providing FICON related information from the Management Server (such as RNID, LIRR, and RLIR data).

For a list of changes to this module by release, refer to the individual module YANG file.

To create a FICON logical switch, see brocade-fibrechannel-logical-switch.

Module Tree

This is the tree view of the module from the `brocade-ficon.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Typesfor data type descriptions.

```
module: brocade-ficon
  +--rw brocade-ficon
     +--rw cup {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw fmsmode-enabled?                                  boolean
     |  +--rw programmed-offline-state-control?                 boolean
     |  +--ro user-alert-mode?                                  boolean
     |  +--rw active-equal-saved-mode?                          boolean
     |  +--rw director-clock-alert-mode?                        boolean
     |  +--ro mihpto?                                           uint32
     |  +--rw unsolicited-alert-mode-fru-enabled?               boolean
     |  +--rw unsolicited-alert-mode-hsc-enabled?               boolean
     |  +--rw unsolicited-alert-mode-invalid-attach-enabled?    boolean
     +--rw logical-path* [link-address channel-image-id] {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw link-address            fibrechannel:brocade-hex-string-type
     |  +--rw channel-image-id        fibrechannel:brocade-hex-string-type
     |  +--ro operational-state?      logical-path-operational-states
     |  +--rw reporting-path-state?   string
     +--ro rnid* [link-address] {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro link-address           fibrechannel:brocade-hex-string-type
     |  +--ro format?                fibrechannel:brocade-hex-string-type
     |  +--ro port-type?             fibrechannel:port-type-string-type
     |  +--ro registered-port-wwn?   fibrechannel:wwn-type
     |  +--ro registered-node-wwn?   fibrechannel:wwn-type
     |  +--ro flags?                 fibrechannel:brocade-hex-string-type
     |  +--ro node-parameters?       fibrechannel:brocade-hex-string-type
     |  +--ro type-number?           string
     |  +--ro model-number?          string
     |  +--ro manufacturer?          string
     |  +--ro plant?                 string
     |  +--ro sequence-number?       string
     |  +--ro tag?                   fibrechannel:brocade-hex-string-type
     +--ro switch-rnid* [switch-address] {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro switch-address     fibrechannel:brocade-hex-string-type
     |  +--ro switch-wwn?        fibrechannel:wwn-type
     |  +--ro flags?             fibrechannel:brocade-hex-string-type
     |  +--ro node-parameters?   fibrechannel:brocade-hex-string-type
     |  +--ro type-number?       string
     |  +--ro model-number?      string
     |  +--ro manufacturer?      string
     |  +--ro plant?             string
```

```
|  +--ro sequence-number?   string
|  +--ro tag?               fibrechannel:brocade-hex-string-type
+--ro lirr* [link-address] {fibrechannel:fibrechannel_switch_platform}?
|  +--ro link-address         fibrechannel:brocade-hex-string-type
|  +--ro format?              fibrechannel:brocade-hex-string-type
|  +--ro port-type?           fibrechannel:port-type-string-type
|  +--ro listener-port-wwn?   fibrechannel:wwn-type
|  +--ro switch-port-wwn?     fibrechannel:wwn-type
|  +--ro listener-type?       string
+--ro rlir* [] {fibrechannel:fibrechannel_switch_platform}?
   +--ro link-address?             fibrechannel:brocade-hex-string-type
   +--ro format?                   fibrechannel:brocade-hex-string-type
   +--ro port-type?                fibrechannel:port-type-string-type
   +--ro port-index?               fibrechannel:user-port-number-type
   +--ro incident-count?           uint8
   +--ro time-stamp-format?        uint8
   +--ro time-stamp?               string
   +--ro port-status?              string
   +--ro link-failure-type?        string
   +--ro registered-port-wwn?      fibrechannel:wwn-type
   +--ro registered-node-wwn?      fibrechannel:wwn-type
   +--ro port-flags?               fibrechannel:brocade-hex-string-type
   +--ro port-node-parameters?     fibrechannel:brocade-hex-string-type
   +--ro port-type-number?         string
   +--ro port-model-number?        string
   +--ro port-manufacturer?        string
   +--ro port-plant?               string
   +--ro port-sequence-number?     string
   +--ro port-tag?                 fibrechannel:brocade-hex-string-type
   +--ro switch-port-wwn?          fibrechannel:wwn-type
   +--ro switch-node-wwn?          fibrechannel:wwn-type
   +--ro switch-flags?             fibrechannel:brocade-hex-string-type
   +--ro switch-node-parameters?   fibrechannel:brocade-hex-string-type
   +--ro switch-type-number?       string
   +--ro switch-model-number?      string
   +--ro switch-manufacturer?      string
   +--ro switch-plant?             string
   +--ro switch-sequence-number?   string
   +--ro switch-tag?               fibrechannel:brocade-hex-string-type
```

Examples

This section provides basic examples for using the brocade-ficon module.

Viewing the FICON CUP Configuration

The following example uses the GET request to view the FICON CUP configuration.

Structure

```
GET <base_URI>/running/brocade-ficon/cup?vf-id=<vfid>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-ficon/cup?vf-id=1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <cup>
        <fmsmode-enabled>true</fmsmode-enabled>
        <programmed-offline-state-control>true</programmed-offline-state-control>
        <user-alert-mode>false</user-alert-mode>
        <active-equal-saved-mode>true</active-equal-saved-mode>
        <director-clock-alert-mode>true</director-clock-alert-mode>
        <mihpto>180</mihpto>
        <unsolicited-alert-mode-fru-enabled>true</unsolicited-alert-mode-fru-enabled>
        <unsolicited-alert-mode-hsc-enabled>true</unsolicited-alert-mode-hsc-enabled>
        <unsolicited-alert-mode-invalid-attach-enabled>true</unsolicited-alert-mode-invalid-attach-enabled>
    </cup>
</Response>
```

Enabling FMS Mode

Before you enable FMS mode, you must be connected to a FICON logical switch. For instructions, refer to the *Brocade Fabric OS FICON Configuration Guide*.

> **NOTE**
> You cannot disable FMS mode through the REST API interface. If a PATCH request is received with fmsmode-enabled=false, an error is returned.

The following example uses the PATCH request to enable FMS mode.

Structure

```
PATCH <base_URI>/running/brocade-ficon/cup?vf-id=<vfid>
```

URI

```
PATCH  https://10.10.10.10/rest/running/brocade-ficon/cup?vf-id=1
```

Request Body

```
<cup>
    <fmsmode-enabled>true</fmsmode-enabled>
</cup>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Configuring FICON CUP

Before you configure FICON CUP, complete the following:

- Create a FICON logical switch (see brocade-fibrechannel-logical-switch)
- Configure FICON (refer to the *Brocade Fabric OS FICON Configuration Guide*)

You can configure the following CUP parameters:

- Allow or prohibit programmed offline state control
- Enable or disable Active=Saved mode
- Enable or disable Director Clock Alert mode
- Enable or disable the unsolicited alert types (HSC, FRU, and invalid attach)

The following example uses the PATCH request to enable offline state control, Active=Saved mode, and Director Clock Alert mode as well as enable the unsolicited alert types.

Structure

```
PATCH <base_URI>/running/brocade-ficon/?vf-id=<vfid>
```

URI

```
PATCH  https://10.10.10.10/rest/running/brocade-ficon/cup?vf-id=1
```

Request Body

```
<cup>
    <programmed-offline-state-control>true</programmed-offline-state-control>
    <active-equal-saved-mode>true</active-equal-saved-mode>
    <director-clock-alert-mode>true</director-clock-alert-mode>
    <unsolicited-alert-mode-fru-enabled>true</unsolicited-alert-mode-fru-enabled>
    <unsolicited-alert-mode-hsc-enabled>true</unsolicited-alert-mode-hsc-enabled>
    <unsolicited-alert-mode-invalid-attach-enabled>true</unsolicited-alert-mode-invalid-attach-enabled>
</cup>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Viewing the FICON Logical Paths

The following example uses the GET request to view the list of logical paths that are currently established to the Control Unit Port (CUP)

Structure

```
GET <base_URI>/running/brocade-ficon/logical-path?vf-id=<vfid>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-ficon/logical-path?vf-id=1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
{
    "Response": {
        "logical-path": [
            {
                "link-address": "0x012b",
                "channel-image-id": "0x03",
                "operational-state": "Oper",
                "reporting-path-state": "Alternate"
            },
            {
                "link-address": "0x012a",
                "channel-image-id": "0x03",
                "operational-state": "Oper",
                "reporting-path-state": "Current"
            },
            {
                "link-address": "0x0129",
                "channel-image-id": "0x03",
                "operational-state": "Reset",
                "reporting-path-state": ""
            },
            {
                "link-address": "0x0128",
                "channel-image-id": "0x03",
                "operational-state": "Oper",
                "reporting-path-state": ""
            }
        ]
    }
}
```

Configuring a FICON Logical Path

The following example uses the PATCH request to set the current reporting path using link address 0x0128 and channel image id 0x03.

Structure

```
PATCH <base_URI>/running/brocade-ficon/logical-path/link-address/link_address/channel-image-
id/channel_image_id/reporting-path-state/current?vf-id=<vfid>
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-ficon/logical-path/link-address/0x0128/channel-image-id/0x03/
reporting-path-state/current?vf-id=32
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content" status appears in the header.

Viewing Request Node Identifier (RNID) Data

The following example uses the GET request to view the RNID information for VFID 1.

Structure

```
GET <base_URI>/running/brocade-ficon/rnid?vf-id=<vfid>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-ficon/rnid?vf-id=1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <rnid>
        <format>0x18</format>
        <port-type>n-port</port-type>
        <link-address>0x0128</link-address>
        <registered-port-wwn>c0:50:76:d5:5f:08:11:81</registered-port-wwn>
        <registered-node-wwn>50:05:07:64:00:cf:c8:87</registered-node-wwn>
        <flags>0x10</flags>
        <node-parameters>0x31014c</node-parameters>
        <type-number>003906</type-number>
        <model-number>M02</model-number>
        <manufacturer>IBM</manufacturer>
        <plant>02</plant>
        <sequence-number>0000000FC887</sequence-number>
        <tag>0xc04c</tag>
    </rnid>
...
    <rnid>
        <format>0x18</format>
        <port-type>n-port</port-type>
        <link-address>0x012b</link-address>
        <registered-port-wwn>c0:50:76:d5:5f:08:11:d1</registered-port-wwn>
        <registered-node-wwn>50:05:07:64:00:cf:c8:87</registered-node-wwn>
        <flags>0x10</flags>
        <node-parameters>0x31014f</node-parameters>
        <type-number>003906</type-number>
        <model-number>M02</model-number>
        <manufacturer>IBM</manufacturer>
        <plant>02</plant>
        <sequence-number>0000000FC887</sequence-number>
        <tag>0xc04f</tag>
```

```
    </rnid>
</Response>
```

Viewing Switch RNID Data

The following example uses the GET request to view the switch RNID information for VFID 1.

Structure

```
GET <base_URI>/running/brocade-ficon/switch-rnid?vf-id=<vfid>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-ficon/switch-rnid?vf-id=1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <switch-rnid>
        <switch-address>0x01</switch-address>
        <switch-wwn>10:00:c4:f5:7c:01:2a:21</switch-wwn>
        <flags>0x00</flags>
        <node-parameters>0x200a00</node-parameters>
        <type-number>SLKWRM</type-number>
        <model-number>630</model-number>
        <manufacturer>BRD</manufacturer>
        <plant>CA</plant>
        <sequence-number>1FKF0324N005</sequence-number>
        <tag>0x01ff</tag>
    </switch-rnid>
</Response>
```

Viewing Link Incident Registration Record (LIRR) Data

The following example uses the GET request to view the LIRR information for VFID 1.

Structure

```
GET <base_URI>/running/brocade-ficon/lirr?vf-id=<vfid>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-ficon/lirr?vf-id=1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <lirr>
        <format>0x18</format>
        <port-type>n-port</port-type>
        <link-address>0x0128</link-address>
        <listener-port-wwn>c0:50:76:d5:5f:08:11:81</listener-port-wwn>
        <switch-port-wwn>20:28:c4:f5:7c:01:2a:21</switch-port-wwn>
        <listener-type>Conditional-C</listener-type>
    </lirr>
    <lirr>
        <format>0x18</format>
        <port-type>n-port</port-type>
        <link-address>0x0129</link-address>
        <listener-port-wwn>c0:50:76:d5:5f:08:11:91</listener-port-wwn>
        <switch-port-wwn>20:29:c4:f5:7c:01:2a:21</switch-port-wwn>
        <listener-type>Conditional-S</listener-type>
    </lirr>
    <lirr>
        <format>0x18</format>
        <port-type>n-port</port-type>
        <link-address>0x012a</link-address>
        <listener-port-wwn>c0:50:76:d5:5f:08:11:c1</listener-port-wwn>
        <switch-port-wwn>20:2a:c4:f5:7c:01:2a:21</switch-port-wwn>
        <listener-type>Conditional</listener-type>
    </lirr>
    <lirr>
        <format>0x18</format>
        <port-type>n-port</port-type>
        <link-address>0x012b</link-address>
        <listener-port-wwn>c0:50:76:d5:5f:08:11:d1</listener-port-wwn>
        <switch-port-wwn>20:2b:c4:f5:7c:01:2a:21</switch-port-wwn>
        <listener-type>Conditional</listener-type>
    </lirr>
</Response>
```

Viewing Registered Link Incident Record (RLIR) Data

The following example uses the GET request to view the RLIR information for VFID 1.

Structure

```
GET <base_URI>/running/brocade-ficon/rlir?vf-id=<vfid>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-ficon/rlir?vf-id=1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <rlir>
        <format>0x18</format>
        <port-type>f-port</port-type>
        <link-address>0x012b</link-address>
        <port-index>43</port-index>
        <incident-count>1</incident-count>
        <time-stamp-format>1</time-stamp-format>
        <time-stamp>Tue May 21 16:42:46 2019</time-stamp>
        <port-status>Link not operational</port-status>
        <link-failure-type>Not operational seq. recognized</link-failure-type>
        <registered-port-wwn>c0:50:76:d5:5f:08:11:d1</registered-port-wwn>
        <registered-node-wwn>50:05:07:64:00:cf:c8:87</registered-node-wwn>
        <port-flags>0x10</port-flags>
        <port-node-parameters>0x31014f</port-node-parameters>
        <port-type-number>003906</port-type-number>
        <port-model-number>M02</port-model-number>
        <port-manufacturer>IBM</port-manufacturer>
        <port-plant>02</port-plant>
        <port-sequence-number>0000000FC887</port-sequence-number>
        <port-tag>0xc04f</port-tag>
        <switch-port-wwn>20:2b:c4:f5:7c:01:2a:21</switch-port-wwn>
        <switch-node-wwn>10:00:c4:f5:7c:01:2a:21</switch-node-wwn>
        <switch-flags>0x00</switch-flags>
        <switch-node-parameters>0x200a2b</switch-node-parameters>
        <switch-type-number>SLKWRM</switch-type-number>
        <switch-model-number>630</switch-model-number>
        <switch-manufacturer>BRD</switch-manufacturer>
        <switch-plant>CA</switch-plant>
        <switch-sequence-number>1FKF0324N005</switch-sequence-number>
        <switch-tag>0x002b</switch-tag>
    </rlir>
</Response>
```

brocade-firmware

This module enables you to manage the firmware version and the integrity of the software image.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-firmware.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-firmware
  +--rw brocade-firmware
     +--ro firmware-history* []
     |  +--ro sequence-number?    uint16
     |  +--ro time-stamp?         yang:date-and-time
     |  +--ro switch-name?        string
     |  +--ro slot-number?        uint16
     |  +--ro process-id?         uint32
     |  +--ro firmware-version?   string
     +--rw firmware-config
        +--rw firmware-check-enabled-state?   boolean
```

Examples

This section provides basic examples for using the brocade-firmware module.

Viewing the Firmware Version on a Device

This example uses a GET request to view the firmware versions.

Structure

```
GET <base_URI>/running/brocade-firmware/firmware-history
```

URI

```
GET 10.10.10.10/rest/running/brocade-firmware/firmware-history
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <firmware-history>
        <sequence-number>1</sequence-number>
        <time-stamp>Mon Mar  1 21:11:55 2021</time-stamp>
        <switch-name>unknown(0)</switch-name>
        <slot-number>2</slot-number>
        <process-id>2063</process-id>
        <firmware-version>Fabos Version 9.1.0fos_major_210226_015811</firmware-version>
    </firmware-history>
```

```
    <firmware-history>
        <sequence-number>2</sequence-number>
        <time-stamp>Tue Mar  2 12:02:55 2021</time-stamp>
        <switch-name>V21SW</switch-name>
        <slot-number>2</slot-number>
        <process-id>2197</process-id>
        <firmware-version>Fabos Version 9.0.0v9.0.0x_patch_210224_164440</firmware-version>
    </firmware-history>
    <firmware-history>
        <sequence-number>3</sequence-number>
        <time-stamp>Tue Mar  2 12:33:35 2021</time-stamp>
        <switch-name>V21SW</switch-name>
        <slot-number>2</slot-number>
        <process-id>2049</process-id>
        <firmware-version>Fabos Version 8.2.2v8.2.2x_patch_210202_040626</firmware-version>
    </firmware-history>
    <firmware-history>
        <sequence-number>4</sequence-number>
        <time-stamp>Thu Mar  4 00:41:27 2021</time-stamp>
        <switch-name>V21SW</switch-name>
        <slot-number>2</slot-number>
        <process-id>2226</process-id>
        <firmware-version>Fabos Version 9.0.1a_bld24</firmware-version>
    </firmware-history>
    <firmware-history>
        <sequence-number>5</sequence-number>
        <time-stamp>Thu Mar  4 09:26:00 2021</time-stamp>
        <switch-name>V21SW</switch-name>
        <slot-number>2</slot-number>
        <process-id>2294</process-id>
        <firmware-version>Fabos Version 9.1.0FWHlatest</firmware-version>
    </firmware-history>
</Response>
```

Checking the Firmware Integrity Check Status

This example uses a GET request to view the firmware integrity check status on the switch.

Structure

```
GET <base_URI>/running/brocade-firmware/firmware-config/firmware-check-enabled-state
```

URI

```
GET 10.10.10.10/rest/running/brocade-firmware/firmware-config/firmware-check-enabled-state
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
```

```
<Response>
    <firmware-config>
        <firmware-check-enabled-state>true</firmware-check-enabled-state>
    </firmware-config>
</Response>
```

Enabling the Firmware Integrity Check

This example uses a PATCH request to enable the firmware integrity check on every reboot of a switch.

Structure

```
PATCH <base_URI>/running/brocade-firmware/firmware-config/firmware-check-enabled-state/true|false
```

URI

```
PATCH 10.10.10.10/rest/running/brocade-firmware/firmware-config/firmware-check-enabled-state/true
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response contains an empty message body, and a “204 No Content” status appears in the header.

To disable the firmware integrity check on every reboot of a switch, set the enabled state to false (`firmware-check-enabled-state/false`).

brocade-fru

This module provides a detailed view of configuration and runtime information of the field replaceable units (FRU) installed in the chassis.

This module also provides information on installed blade, fan, and power-supply FRUs. In addition, customer-supplied information, if included, can be viewed for each FRU type.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-fru.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Typesfor data type descriptions.

```
module: brocade-fru
  +--rw brocade-fru
     +--rw blade* [slot-number]
     |  +--rw slot-number                  uint16
     |  +--ro manufacturer?                string
     |  +--ro blade-type?                  blade-type-type
     |  +--ro blade-id?                    uint16
     |  +--ro blade-state?                 blade-state-type
     |  +--ro model-name?                  fibrechannel:product-name-type
     |  +--ro firmware-version?            fru:firmware-version-type
     |  +--ro fc-port-count?               uint16
     |  +--ro ge-port-count?               uint16
     |  x--ro ip-address-list
     |  |  +--ro ip-address*   union
     |  +--rw static-ip-addresses
     |  |  +--rw ip-address*   inet:ip-address
     |  +--ro stateful-ip-addresses
     |  |  +--ro ip-address*   union
     |  +--ro stateless-ipv6-addresses
     |  |  +--ro ip-address*   union
     |  +--ro link-local-address?          union
     |  x--ro ip-gateway-list
     |  |  +--ro ip-gateway*   fru:ip-gateway-type
     |  x--ro subnet-mask?                 inet:ipv4-address
     |  +--ro primary-firmware-version?    fru:firmware-version-type
     |  +--ro secondary-firmware-version?  fru:firmware-version-type
     |  +--ro power-consumption?           fru:power-detail-type
     |  +--ro power-usage?                 fru:power-detail-type
     |  +--ro extension-enabled?           boolean
     |  +--rw extension-app-mode?          string
     |  +--rw extension-ve-mode?           string
     |  +--rw extension-ge-mode?           string
     |  +--ro manufacture-date?            fru:fru-date-type {fibrechannel:fibrechannel_chassis_platform}?
     |  +--ro last-header-update-date?     fru:fru-date-type {fibrechannel:fibrechannel_chassis_platform}?
     |  +--ro header-version?              uint16 {fibrechannel:fibrechannel_chassis_platform}?
     |  +--ro hardware-version?            fibrechannel:brocade-hex-string-type
     |  +--ro time-alive?                  fru:time-detail-type
     |  +--ro time-awake?                  fru:time-detail-type
     |  +--ro part-number?                 fru:part-number-type
```

```
|  +--ro serial-number?               fru:serial-number-type
|  +--ro vendor-serial-number?        string
|  +--ro vendor-part-number?          string
|  +--ro vendor-revision-number?      string
|  +--ro vendor-id?                   string
+--ro fan* [unit-number]
|  +--ro unit-number                uint16
|  +--ro power-consumption?         fru:power-detail-type
|  +--ro operational-state?         string
|  +--ro speed?                     uint32
|  +--ro airflow-direction?         air-flow-direction-type
|  +--ro time-alive?                fru:time-detail-type
|  +--ro time-awake?                fru:time-detail-type
|  +--ro part-number?               fru:part-number-type
|  +--ro serial-number?             fru:serial-number-type
|  +--ro manufacture-date?          fru:fru-date-type
|  +--ro last-header-update-date?   fru:fru-date-type {fibrechannel:fibrechannel_chassis_platform}?
|  +--ro header-version?            uint16
|  +--ro vendor-serial-number?      string
|  +--ro vendor-part-number?        string
|  +--ro vendor-revision-number?    string
|  +--ro vendor-id?                 string
+--ro power-supply* [unit-number]
|  +--ro unit-number                    uint16
|  +--ro power-production?              uint32
|  +--ro input-voltage?                 decimal64
|  +--ro airflow-direction?             air-flow-direction-type
|  +--ro power-source?                  string
|  +--ro operational-state?             string
|  +--ro temperature?                   decimal64
|  +--ro power-usage?                   fru:power-detail-type
|  +--ro temperature-sensor-supported?  boolean
|  +--ro manufacture-date?              fru:fru-date-type
|  +--ro last-header-update-date?       fru:fru-date-type {fibrechannel:fibrechannel_chassis_platform}?
|  +--ro header-version?                uint16
|  +--ro time-alive?                    fru:time-detail-type
|  +--ro time-awake?                    fru:time-detail-type
|  +--ro part-number?                   fru:part-number-type
|  +--ro serial-number?                 fru:serial-number-type
|  +--ro vendor-serial-number?          string
|  +--ro vendor-part-number?            string
|  +--ro vendor-revision-number?        string
|  +--ro vendor-id?                     string
+--ro history-log* []
|  +--ro fru-type?        fru:fru-type
|  +--ro position?        uint16
|  +--ro state?           fru:state-type
|  +--ro time-stamp?      yang:date-and-time
|  +--ro part-number?     fru:part-number-type
|  +--ro serial-number?   fru:serial-number-type
+--ro sensor* [id]
|  +--ro id                          uint16
|  +--ro slot-number?                uint16
```

```
|  +--ro index?                       uint16
|  +--ro state?                       fru:operational-state-type
|  +--ro category?                    fru:sensor-category
|  +--ro temperature?                 uint16
|  +--ro temperature-in-fahrenheit?   int32
+--ro wwn* [unit-number]
   +--ro unit-number                uint16
   +--ro operational-state?         fru:operational-state-type
   +--ro airflow-direction?         air-flow-direction-type
   +--ro power-usage?               fru:power-detail-type
   +--ro generation-number?         uint16 {fibrechannel:fibrechannel_chassis_platform}?
   +--ro manufacture-date?          fru:fru-date-type
   +--ro last-header-update-date?   fru:fru-date-type
   +--ro header-version?            uint16
   +--ro time-alive-in-hours?       fru:time-detail-type-in-hours
   +--ro time-awake-in-hours?       fru:time-detail-type-in-hours
   +--ro part-number?               fru:part-number-type
   +--ro serial-number?             fru:serial-number-type
   +--ro time-alive?                fru:time-detail-type
   +--ro time-awake?                fru:time-detail-type
   +--ro vendor-serial-number?      string
   +--ro vendor-part-number?        string
   +--ro vendor-revision-number?    string
   +--ro vendor-id?                 string
```

Examples

This section provides basic examples for using the brocade-fru module.

Setting the Extension Application Mode Configuration

The following example uses the PATCH request to set the application mode configuration of the Extension blade to FCIP.

Structure

```
PATCH <base_URI>/running/brocade-fru/blade
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fru/blade
```

Request Body

```
<blade>
    <slot-number>4</slot-number>
    <extension-app-mode>FCIP</extension-app-mode>
<blade>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status appears in the header.

Setting the Extension VE Mode Configuration

The following example uses the PATCH request to set the VE mode configuration of the Extension blade to 20VE.

Structure

```
PATCH <base_URI>/running/brocade-fru/blade
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fru/blade
```

Request Body

```
<blade>
    <slot-number>4</slot-number>
    <extension-ve-mode>20VE</extension-ve-mode>
<blade>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status appears in the header.

Setting the Extension GE Mode Configuration

The following example uses the PATCH request to set the GE mode configuration of the Extension blade to optical.

Structure

```
PATCH <base_URI>/running/brocade-fru/blade
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-fru/blade
```

Request Body

```
<blade>
    <slot-number>4</slot-number>
    <extension-ge-mode>optical</extension-ge-mode>
<blade>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status appears in the header.

Viewing Data for all Fans in the Chassis

The following example uses the GET request to view data on all fans in the chassis.

Structure

```
GET <base_URI>/running/brocade-fru/fan
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fru/fan
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fan>
        <unit-number>1</unit-number>
        <serial-number>DYL3017M0BG</serial-number>
        <part-number>60-1003203-04</part-number>
        <power-consumption>-300</power-consumption>
        <operational-state>ok</operational-state>
        <airflow-direction>non-portside intake</airflow-direction>
        <speed>5076</speed>
        <time-alive>435</time-alive>
        <time-awake>2</time-awake>
    </fan>
    <fan>
        <unit-number>2</unit-number>
        <serial-number>DYL3017M06K</serial-number>
        <part-number>60-1003203-04</part-number>
        <power-consumption>-300</power-consumption>
        <operational-state>ok</operational-state>
        <airflow-direction>non-portside intake</airflow-direction>
        <speed>5012</speed>
        <time-alive>435</time-alive>
        <time-awake>2</time-awake>
    </fan>
</Response>
```

Viewing Data for a Specific Fan

The following example uses the GET request to view data for a specific fan (1) in the chassis.

Structure

```
GET <base_URI>/running/brocade-fru/fan/unit-number/fan-unit
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fru/fan/unit-number/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fan>
        <unit-number>1</unit-number>
        <serial-number>DYL3017M0BG</serial-number>
        <part-number>60-1003203-04</part-number>
        <power-consumption>-300</power-consumption>
        <operational-state>ok</operational-state>
        <airflow-direction>non-portside intake</airflow-direction>
        <speed>5076</speed>
        <time-alive>435</time-alive>
        <time-awake>2</time-awake>
    </fan>
</Response>
```

Viewing Data for a Specific Power Supply

The following example uses the GET request to view data on a specific power supply (1) in the chassis.

Structure

```
GET <base_URI>/running/brocade-fru/power-supply/unit-number/power-supply-unit
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fru/power-supply/unit-number/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <power-supply>
        <unit-number>1</unit-number>
        <serial-number>DUC2M32M2FD</serial-number>
        <part-number>23-0000161-01</part-number>
        <power-production>2870</power-production>
        <power-source>AC</power-source>
        <operational-state>ok</operational-state>
        <airflow-direction>non-portside intake</airflow-direction>
        <input-voltage>203.50</input-voltage>
        <temperature>30.00</temperature>
        <time-alive>747</time-alive>
        <time-awake>0</time-awake>
        <temperature-sensor-supported>true</temperature-sensor-supported>
    </power-supply>
</Response>
```

Viewing the History Log

This example uses a GET request to view the history log for all FRUs in the chassis.

Structure

```
GET <base_URI>/running/brocade-fru/history-log
```

URI

```
GET https://10.10.10.10/rest/running/brocade-fru/history-log
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <history-log>
        <fru-type>core blade</fru-type>
        <position>5</position>
        <state>inserted</state>
        <time-stamp>2017-06-14T13:11:31-06:00</time-stamp>
        <serial-number>DZD3213M011</serial-number>
        <part-number>60-1003226-10</part-number>
    </history-log>
    <history-log>
        <fru-type>power supply</fru-type>
        <position>1</position>
```

```
        <state>inserted</state>
        <time-stamp>2017-06-14T13:11:49-06:00</time-stamp>
        <serial-number>DUC2M32M2FD</serial-number>
        <part-number>23-0000161-01</part-number>
    </history-log>
    <history-log>
        <fru-type>power supply</fru-type>
        <position>2</position>
        <state>inserted</state>
        <time-stamp>2017-06-14T13:11:50-06:00</time-stamp>
        <serial-number>DUC2M32M2ME</serial-number>
        <part-number>23-0000161-01</part-number>
    </history-log>
    <history-log>
        <fru-type>fan</fru-type>
        <position>1</position>
        <state>inserted</state>
        <time-stamp>2017-06-14T13:11:50-06:00</time-stamp>
        <serial-number>DYL3017M0BG</serial-number>
        <part-number>60-1003203-04</part-number>
    </history-log>
    <history-log>
        <fru-type>fan</fru-type>
        <position>2</position>
        <state>inserted</state>
        <time-stamp>2017-06-14T13:11:50-06:00</time-stamp>
        <serial-number>DYL3017M06K</serial-number>
        <part-number>60-1003203-04</part-number>
    </history-log>
    <history-log>
        <fru-type>wwn</fru-type>
        <position>1</position>
        <state>inserted</state>
        <time-stamp>2017-06-14T13:11:51-06:00</time-stamp>
        <serial-number>DZH3010M018</serial-number>
        <part-number>60-1003194-02</part-number>
    </history-log>
    <history-log>
        <fru-type>wwn</fru-type>
        <position>2</position>
        <state>inserted</state>
        <time-stamp>2017-06-14T13:11:51-06:00</time-stamp>
        <serial-number>DZH3010M037</serial-number>
        <part-number>60-1003194-02</part-number>
    </history-log>
    <history-log>
        <fru-type>cp blade</fru-type>
        <position>2</position>
        <state>inserted</state>
        <time-stamp>2017-06-14T13:11:52-06:00</time-stamp>
        <serial-number>DYK3213M03G</serial-number>
        <part-number>60-1003201-12</part-number>
    </history-log>
```

```
<history-log>
    <fru-type>sw blade</fru-type>
    <position>8</position>
    <state>inserted</state>
    <time-stamp>2017-08-08T15:24:27-06:00</time-stamp>
    <serial-number>FDU0325N021</serial-number>
    <part-number>60-1003847-01</part-number>
</history-log>
<history-log>
    <fru-type>cp blade</fru-type>
    <position>2</position>
    <state>inserted</state>
    <time-stamp>2017-09-15T09:19:57-06:00</time-stamp>
    <serial-number>DYK3213M03G</serial-number>
    <part-number>60-1003201-12</part-number>
</history-log>
<history-log>
    <fru-type>power supply</fru-type>
    <position>1</position>
    <state>removed</state>
    <time-stamp>2018-03-07T10:24:17-07:00</time-stamp>
    <serial-number>DUC2M32M2FD</serial-number>
    <part-number>23-0000161-01</part-number>
</history-log>
<history-log>
    <fru-type>power supply</fru-type>
    <position>1</position>
    <state>inserted</state>
    <time-stamp>2018-03-27T08:16:02-06:00</time-stamp>
    <serial-number>DUC2M32M2FD</serial-number>
    <part-number>23-0000161-01</part-number>
</history-log>
<history-log>
    <fru-type>core blade</fru-type>
    <position>6</position>
    <state>removed</state>
    <time-stamp>2018-04-13T10:29:46-06:00</time-stamp>
    <serial-number>DZD3221M001</serial-number>
    <part-number>60-1003226-10</part-number>
</history-log>
<history-log>
    <fru-type>core blade</fru-type>
    <position>6</position>
    <state>inserted</state>
    <time-stamp>2018-04-13T10:32:10-06:00</time-stamp>
    <serial-number>DZD3221M001</serial-number>
    <part-number>60-1003226-10</part-number>
</history-log>
<history-log>
    <fru-type>core blade</fru-type>
    <position>5</position>
    <state>removed</state>
    <time-stamp>2018-04-13T10:32:57-06:00</time-stamp>
```

```
        <serial-number>DZD3213M011</serial-number>
        <part-number>60-1003226-10</part-number>
    </history-log>
    <history-log>
        <fru-type>sw blade</fru-type>
        <position>7</position>
        <state>inserted</state>
        <time-stamp>2018-04-13T10:33:20-06:00</time-stamp>
        <serial-number>DYJ3232N09B</serial-number>
        <part-number>60-1003200-12</part-number>
    </history-log>
    <history-log>
        <fru-type>core blade</fru-type>
        <position>5</position>
        <state>inserted</state>
        <time-stamp>2018-04-13T10:33:47-06:00</time-stamp>
        <serial-number>DZD3213M011</serial-number>
        <part-number>60-1003226-10</part-number>
    </history-log>
    <history-log>
        <fru-type>sw blade</fru-type>
        <position>8</position>
        <state>removed</state>
        <time-stamp>2019-01-04T09:49:37-07:00</time-stamp>
        <serial-number>FDU0325N021</serial-number>
        <part-number>60-1003847-01</part-number>
    </history-log>
    <history-log>
        <fru-type>sw blade</fru-type>
        <position>8</position>
        <state>inserted</state>
        <time-stamp>2019-01-04T10:53:43-07:00</time-stamp>
        <serial-number>FDU0325N021</serial-number>
        <part-number>60-1003847-01</part-number>
    </history-log>
</Response>
```

brocade-interface/extension-ip-interface

The extension IP interface parameters in this module are used to retrieve or configure IP interfaces on the specified switch.

For a list of changes to this module by release, refer to the individual module YANG file.

> **NOTE**
> The brocade-fibrechannel, brocade-extension-ip-interface, and brocade-gigabitethernet modules have been merged into the brocade-interface module in Fabric OS 8.2.1b or later. However, this section only covers brocade-extension-ip-interface. For information about brocade-fibrechannel, refer to brocade-interface/fibrechannel. For information about brocade-gigabitethernet, refer to brocade-interface/gigabitethernet.

Module Tree

This is the tree view of the module from the brocade-extension-ip-interface.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-interface
  +--rw brocade-interface
     +--rw extension-ip-interface* [name dp-id ip-address] {fibrechannel:fibrechannel_extension_platform}?
     |  +--rw name                brocade-interface-types:ip-extension-interface-type
     |  +--rw dp-id               uint16
     |  +--rw ip-address          union
     |  +--rw ip-prefix-length    brocade-interface-types:ip-prefix-length
     |  +--rw mtu-size?           uint16
     |  +--rw vlan-id?            uint16
     |  +--ro status-flags?       string
```

Examples

This section provides basic examples for using the brocade-interface/extension-ip-interface module.

Creating an IP Interface

This example uses a POST request to create an IP interface on port 4/5 with a MTU of 1500 and a VLAN ID of 100.

Structure

```
POST <base_URI>/running/brocade-interface/extension-ip-interface
```

URI

```
POST http://10.10.10.40/rest/running/brocade-interface/extension-ip-interface
```

Request Body

Notice that the slash character in the name resource is *not* encoded as “%2f”. This is because it is in the request body.

```
<extension-ip-interface>
   <name>4/5</name>
   <dp-id>0</dp-id>
   <ip-address>10.10.10.1</ip-address>
   <mtu-size>1500</mtu-size>
   <vlan-id>100</vlan-id>
</extension-ip-interface>
```

Response Body

When the resource is successfully created, the response contains an empty message body and a “201 Created” status in the headers.

Deleting an IP Interface

This example uses a DELETE request to delete the IP interface on port 4/5.

Structure

```
DELETE <base_URI>/running/brocade-interface/extension-ip-interface
```

URI

```
DELETE http://10.10.10.10/rest/running/brocade-interface/extension-ip-interface
```

Request Body

```
<extension-ip-interface>
   <name>4/5</name>
   <dp-id>0</dp-id>
   <ip-address>10.10.10.10</ip-address>
   <ip-prefix-length>24</ip-prefix-length>
</extension-ip-interface>
```

Response Body

When the resource is successfully deleted, the response contains an empty message body and a “204 No Content” status in the headers.

Retrieving Information on an IP Interface

The following example uses a GET request to retrieve information on the IP interface on port 0/2.

Structure

```
GET <base_URI>/running/brocade-interface/extension-ip-interface
```

URI

```
GET https://10.10.10.10/rest/running/brocade-interface/extension-ip-interface/name/0%2f2/dp-id/0/ip-
address/192.168.20.1/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
      <extension-ip-interface>
      <name>0/2</name>
      <dp-id>0</dp-id>
      <ip-address>192.168.20.1</ip-address>
```

```
      <ip-prefix-length>24</ip-prefix-length>
      <mtu-size>1500</mtu-size>
      <vlan-id>0</vlan-id>
      <status-flags>U R M I </status-flags>
   </extension-ip-interface>
</Response>
```

brocade-interface/fibrechannel

The Fibre Channel parameters in this module retrieve information for all Fibre Channel ports on the specified switch. If Virtual Fabrics are enabled, the request can include a query parameter (vf-id) for the desired virtual fabric. If no query parameter is specified and Virtual Fabrics are enabled, the port information for the switch in the default virtual fabric is returned.

For a list of changes to this module by release, refer to the individual module YANG file.

> **NOTE**
> The brocade-fibrechannel, brocade-extension-ip-interface, and brocade-gigabitethernet modules have been merged into the brocade-interface module in Fabric OS 8.2.1b and later. However, this section covers only brocade-fibrechannel. For information about brocade-extension-ip-interface, see brocade-interface/extension-ip-interface. For information about brocade-gigabitethernet, see brocade-interface/gigabitethernet.

This module augments the interface and interface-state lists defined in the ietf-interfaces module as defined in RFC 7223 with Fibre Channel data nodes, and it adds Fibre Channel state data.

Module Tree

This is the tree view of the module from the `brocade-interface.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-interface
  +--rw brocade-interface
     +--rw fibrechannel* [name]
     |  +--rw name                                          fibrechannel:slot-port-name-type
     |  +--ro wwn?                                          fibrechannel:wwn-type
     |  x--ro operational-status?                           uint32
     |  +--ro operational-status-string?                    operational-status-type
     |  x--rw enabled-state?                                uint32
     |  +--rw is-enabled-state?                             boolean
     |  +--rw user-friendly-name?                           string
     |  x--rw speed?                                        brocade-interface-types:fc-speed-type
     |  +--rw protocol-speed?                               fibrechannel:protocol-speed-type
     |  x--ro max-speed?                                    brocade-interface-types:fc-speed-type
     |  +--ro max-protocol-speed?                           fibrechannel:protocol-speed-type
     |  x--ro auto-negotiate?                               uint8
     |  +--ro auto-negotiate-v2?                            boolean
     |  o--rw g-port-locked?                                uint8
  {fibrechannel:fibrechannel_switch_platform}?
     |  x--rw e-port-disable?                               uint8
  {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw e-port-disable-v2?                            boolean
  {fibrechannel:fibrechannel_switch_platform}?
     |  x--rw n-port-enabled?                               uint8
  {fibrechannel:access_gateway_platform}?
     |  +--rw n-port-enabled-v2?                            boolean
  {fibrechannel:access_gateway_platform}?
     |  x--rw d-port-enable?                                uint8
     |  +--rw d-port-enable-v2?                             boolean
     |  x--rw persistent-disable?                           uint8
     |  +--rw persistent-disable-v2?                        boolean
     |  +--rw application-header-enabled?                   boolean
  {fibrechannel:fibrechannel_switch_platform}?
```

```
      |  +--ro locked-e-port-enabled?                              boolean
  {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro dwdm-losync-enabled?                                boolean
  {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro d-port-dwdm-enabled?                                boolean
  {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro neighbor
      |  |  +--ro wwn*   fibrechannel:wwn-type
      |  +--ro neighbor-slot-port?                                 fibrechannel:slot-port-name-type
      |  +--ro neighbor-node-wwn?                                  union
      |  +--ro neighbor-port-index?                                fibrechannel:user-port-number-type
      |  +--ro neighbor-switch-user-friendly-name?                 string
      |  x--ro fcid?                                               fibrechannel:fcid-type
      |  +--ro fcid-hex?                                           fibrechannel:fcid-hex-string-type
      |  x--ro port-type?                                          brocade-interface-types:port-type-type
      |  +--ro port-type-string?                                   brocade-interface-types:port-type-string-
 type
      |  +--ro icl-interconnect-status?                            boolean
      |  x--rw qos-enabled?                                        uint8
      |  +--rw qos-enabled-v2?                                     boolean
      |  x--rw compression-configured?                             uint8
  {fibrechannel:fibrechannel_switch_platform}?
      |  +--rw compression-configured-v2?                          boolean
  {fibrechannel:fibrechannel_switch_platform}?
      |  x--rw encryption-enabled?                                 uint8
  {fibrechannel:fibrechannel_switch_platform}?
      |  +--rw encryption-enabled-v2?                              boolean
  {fibrechannel:fibrechannel_switch_platform}?
      |  x--ro compression-active?                                 uint8
  {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro compression-active-v2?                              boolean
  {fibrechannel:fibrechannel_switch_platform}?
      |  x--ro encryption-active?                                  uint8
  {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro encryption-active-v2?                              boolean
  {fibrechannel:fibrechannel_switch_platform}?
      |  x--rw target-driven-zoning-enable?                        uint8
      |  +--rw target-driven-zoning-enable-v2?                     boolean
      |  x--rw sim-port-enabled?                                   uint8
  {fibrechannel:fibrechannel_switch_platform}?
      |  +--rw sim-port-enabled-v2?                                boolean
  {fibrechannel:fibrechannel_switch_platform}?
      |  x--rw mirror-port-enabled?                                int8
      |  +--rw mirror-port-enabled-v2?                             boolean
      |  x--rw credit-recovery-enabled?                            int8
      |  +--rw credit-recovery-enabled-v2?                         boolean
      |  x--ro credit-recovery-active?                             int8
      |  +--ro credit-recovery-active-v2?                          boolean
      |  x--ro fec-active?                                         int8
      |  +--ro fec-active-v2?                                      boolean
      |  +--rw f-port-buffers?                                     uint8
      |  +--rw e-port-credit?                                      uint8
  {fibrechannel:fibrechannel_switch_platform}?
```

```
    |  x--rw csctl-mode-enabled?                                  int8
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw csctl-mode-enabled-v2?                               boolean
{fibrechannel:fibrechannel_switch_platform}?
    |  x--rw fault-delay-enabled?                                 uint8
    |  +--rw fault-delay-enabled-v2?                              boolean
    |  x--rw octet-speed-combo?                                   int8
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw octet-speed-combo-string?                            octet-speed-combo-type
{fibrechannel:fibrechannel_switch_platform}?
    |  x--ro long-distance?                                       uint8
{fibrechannel:fibrechannel_switch_platform}?
    |  +--ro long-distance-string?                               long-distance-type
{fibrechannel:fibrechannel_switch_platform}?
    |  x--ro vc-link-init?                                        uint8
{fibrechannel:fibrechannel_switch_platform}?
    |  +--ro vc-link-init-v2?                                    boolean
{fibrechannel:fibrechannel_switch_platform}?
    |  x--rw isl-ready-mode-enabled?                             uint8
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw isl-ready-mode-enabled-v2?                          boolean
{fibrechannel:fibrechannel_switch_platform}?
    |  x--rw rscn-suppression-enabled?                           uint8
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw rscn-suppression-enabled-v2?                        boolean
{fibrechannel:fibrechannel_switch_platform}?
    |  x--rw los-tov-mode-enabled?                               uint8
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw los-tov-mode-enabled-string?                        los-tov-mode-type
{fibrechannel:fibrechannel_switch_platform}?
    |  x--rw npiv-enabled?                                       uint8
    |  +--rw npiv-enabled-v2?                                    boolean
    |  +--rw npiv-pp-limit?                                      int16
    |  x--ro npiv-flogi-logout-enabled?                          int8
    |  +--ro npiv-flogi-logout-enabled-v2?                       boolean
    |  x--rw ex-port-enabled?                                    uint8
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw ex-port-enabled-v2?                                 boolean
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw fc-router-port-cost?                                uint32
    |  +--rw edge-fabric-id?                                     fibrechannel:fabric-id-type
    |  +--ro destination-fabric-principal-switch-wwn?            fibrechannel:wwn-type
    |  +--rw preferred-front-domain-id?                          fibrechannel:domain-id-type
    |  +--ro neighbor-switch-ipv4-address?                       union
    |  +--ro neighbor-switch-ipv6-address?                       union
    |  +--ro neighbor-fabric-name?                               string
    |  x--rw fec-enabled?                                        int8
    |  +--rw fec-enabled-v2?                                     boolean
    |  x--ro via-tts-fec-enabled?                                int8
    |  +--ro via-tts-fec-enabled-v2?                             boolean
    |  x--rw port-autodisable-enabled?                           uint8
    |  +--rw port-autodisable-enabled-v2?                        boolean
```

```
    |  o--rw rate-limit-enabled?                                    int16
{fibrechannel:fibrechannel_switch_platform}?
    |  o--rw non-dfe-enabled?                                       uint8
{fibrechannel:fibrechannel_switch_platform}?
    |  x--rw trunk-port-enabled?                                    uint8
    |  +--rw trunk-port-enabled-v2?                                 boolean
    |  +--ro default-index?                                         fibrechannel:user-port-number-type
    |  +--ro physical-state?                                        string
    |  x--ro pod-license-status?                                    boolean
    |  +--rw pod-license-state?                                     enumeration
{fibrechannel:fibrechannel_standalone_switch_platform}?
    |  +--ro index?                                                 fibrechannel:user-port-number-type
    |  +--ro reserved-buffers?                                      uint16
    |  +--ro average-transmit-buffer-usage?                         uint16
    |  +--ro average-transmit-frame-size?                           uint16
    |  +--ro average-receive-buffer-usage?                          uint16
    |  +--ro average-receive-frame-size?                            uint16
    |  +--ro current-buffer-usage?                                  uint16
    |  +--ro recommended-buffers?                                   uint16
    |  +--ro measured-link-distance?                                string
    |  +--ro chip-instance?                                         uint16
    |  +--ro chip-buffers-available?                                uint16
    |  +--ro port-health?                                           string
    |  +--ro authentication-protocol?                               string
    |  +--rw disable-reason?                                        string
    |  +--ro areas
    |  |  +--ro area*   fibrechannel:area-address-type
    |  +--ro le-domain?                                             fibrechannel:domain-id-type
    |  +--rw port-peer-beacon-enabled?                              boolean
    |  +--rw clean-address-enabled?                                 boolean
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw congestion-signal-enabled?                             boolean
    |  +--ro segmentation-reason?                                   string
    |  +--ro lag-name?                                              string
    |  +--ro lag-member-link-status?                                fc-lag-link-status-type
    |  +--ro bound-address-list {fibrechannel:fibrechannel_switch_platform}?
    |  |  +--ro bound-address*   fibrechannel:area-address-type
    |  +--ro user-bound-enabled?                                    boolean
{fibrechannel:fibrechannel_switch_platform}?
    |  +--ro address-binding-mode?                                  port-address-binding-mode
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw ms-acl-application-server-access?                      brocade-interface-types:ms-acl-access-type
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw ms-acl-enhanced-fabric-configuration-server-access?    brocade-interface-types:ms-acl-access-type
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw ms-acl-fabric-configuration-server-access?             brocade-interface-types:ms-acl-access-type
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw ms-acl-fabric-device-management-interface-access?      brocade-interface-types:ms-acl-access-type
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw ms-acl-fabric-zone-server-access?                      brocade-interface-types:ms-acl-access-type
{fibrechannel:fibrechannel_switch_platform}?
    |  +--rw ms-acl-unzoned-name-server-access?                     brocade-interface-types:ms-acl-access-type
{fibrechannel:fibrechannel_switch_platform}?
```

```
|  +--ro port-generation-number?                          uint32
|  +--ro port-scn?                                        port-scn-type
+--rw fibrechannel-statistics* [name]
|  +--rw name                                            fibrechannel:slot-port-name-type
|  +--ro sampling-interval?                              uint16
|  +--ro time-generated?                                 fibrechannel:time-generated-type
|  +--ro time-refreshed?                                 fibrechannel:time-generated-type
|  x--rw reset-statistics?                               uint8
|  +--ro in-octets?                                      yang:zero-based-counter64
|  +--ro out-octets?                                     yang:zero-based-counter64
|  +--ro in-multicast-pkts?                              yang:zero-based-counter64
|  +--ro out-multicast-pkts?                             yang:zero-based-counter64
|  +--ro in-link-resets?                                 yang:zero-based-counter64
|  +--ro out-link-resets?                                yang:zero-based-counter64
|  +--ro in-offline-sequences?                           yang:zero-based-counter64
|  +--ro out-offline-sequences?                          yang:zero-based-counter64
|  +--ro invalid-ordered-sets?                           yang:zero-based-counter64
|  +--ro frames-too-long?                                yang:zero-based-counter64
|  +--ro truncated-frames?                               yang:zero-based-counter64
|  +--ro address-errors?                                 yang:zero-based-counter64
|  +--ro delimiter-errors?                               yang:zero-based-counter64
|  +--ro encoding-disparity-errors?                      yang:zero-based-counter64
|  +--ro too-many-rdys?                                  yang:zero-based-counter64
|  +--ro in-crc-errors?                                  yang:zero-based-counter64
|  +--ro crc-errors?                                     yang:zero-based-counter64
|  +--ro bad-eofs-received?                              yang:zero-based-counter64
|  +--ro encoding-errors-outside-frame?                  yang:zero-based-counter64
|  +--ro multicast-timeouts?                             yang:zero-based-counter64
|  +--ro in-lcs?                                         yang:zero-based-counter64
|  +--ro in-frame-rate?                                  yang:zero-based-counter64
|  +--ro out-frame-rate?                                 yang:zero-based-counter64
|  +--ro in-max-frame-rate?                              yang:zero-based-counter64
|  +--ro out-max-frame-rate?                             yang:zero-based-counter64
|  +--ro in-rate?                                        yang:zero-based-counter64
|  +--ro out-rate?                                       yang:zero-based-counter64
|  +--ro in-peak-rate?                                   yang:zero-based-counter64
|  +--ro out-peak-rate?                                  yang:zero-based-counter64
|  +--ro in-frames?                                      yang:zero-based-counter64
|  +--ro out-frames?                                     yang:zero-based-counter64
|  +--ro bb-credit-zero?                                 yang:zero-based-counter64
|  +--ro input-buffer-full?                              yang:zero-based-counter64
|  +--ro f-busy-frames?                                  yang:zero-based-counter64
|  +--ro p-busy-frames?                                  yang:zero-based-counter64
|  +--ro f-rjt-frames?                                   yang:zero-based-counter64
|  +--ro p-rjt-frames?                                   yang:zero-based-counter64
|  +--ro class-1-frames?                                 yang:zero-based-counter64
|  +--ro class-2-frames?                                 yang:zero-based-counter64
|  +--ro class-3-frames?                                 yang:zero-based-counter64
|  +--ro class-3-discards?                               yang:zero-based-counter64
|  +--ro link-failures?                                  yang:zero-based-counter64
|  +--ro invalid-transmission-words?                     yang:zero-based-counter64
|  +--ro primitive-sequence-protocol-error?              yang:zero-based-counter64
|  +--ro loss-of-signal?                                 yang:zero-based-counter64
```

```
|  +--ro loss-of-sync?                              yang:zero-based-counter64
|  +--ro class3-in-discards?                        yang:zero-based-counter64
|  +--ro class3-out-discards?                       yang:zero-based-counter64
|  +--ro pcs-block-errors?                          yang:zero-based-counter64
|  +--ro remote-link-failures?                      yang:zero-based-counter64
|  +--ro remote-invalid-transmission-words?         yang:zero-based-counter64
|  +--ro remote-primitive-sequence-protocol-error?  yang:zero-based-counter64
|  +--ro remote-loss-of-signal?                     yang:zero-based-counter64
|  +--ro remote-loss-of-sync?                       yang:zero-based-counter64
|  +--ro remote-crc-errors?                         yang:zero-based-counter64
|  +--ro remote-fec-uncorrected?                    yang:zero-based-counter64
|  +--ro remote-buffer-credit-info
|  |  +--ro bb-credit?        yang:zero-based-counter64
|  |  +--ro peer-bb-credit?   yang:zero-based-counter64
|  +--ro link-level-interrupts?                     yang:zero-based-counter64
|  +--ro frames-processing-required?                yang:zero-based-counter64
|  +--ro frames-timed-out?                          yang:zero-based-counter64
|  +--ro frames-transmitter-unavailable-errors?     yang:zero-based-counter64
|  +--ro non-operational-sequences-in?              yang:zero-based-counter64
|  +--ro non-operational-sequences-out?             yang:zero-based-counter64
|  +--ro fec-uncorrected?                           yang:zero-based-counter64
|  +--ro total-up-time?                             decimal64
|  +--ro total-down-time?                           decimal64
|  +--ro down-occurrence?                           yang:zero-based-counter32
|  +--ro total-offline-time?                        decimal64
|  +--ro encoding-error-in?                         yang:zero-based-counter64
|  +--ro tx-peak-frame?                             yang:zero-based-counter64
|  +--ro rx-peak-frame?                             yang:zero-based-counter64
|  +--ro starvation-tenancy-stop?                   yang:zero-based-counter64
|  +--ro invalid-arbitrate-loop?                    yang:zero-based-counter64
|  +--ro ftb-type-1-miss?                           yang:zero-based-counter64
|  +--ro ftb-type-2-miss?                           yang:zero-based-counter64
|  +--ro ftb-type-6-miss?                           yang:zero-based-counter64
|  +--ro hard-zone-miss?                            yang:zero-based-counter64
|  +--ro lun-zone-miss?                             yang:zero-based-counter64
|  +--ro state-transition-count?                    uint32
|  +--ro interrupt-count?                           uint32
|  +--ro unknown-interrupt-count?                   uint32
|  +--ro congestion-primitive-in?                   uint32
+--ro fibrechannel-router-statistics* [port-index] {fibrechannel:fibrechannel_router_platform}?
|  +--ro port-index            fibrechannel:port-number-type
|  +--ro proxy-device-count?   uint32
|  +--ro nr-port-count?        uint32
```

Examples

This section provides basic examples for using the brocade-interface/fibrechannel module.

Comparable Fabric OS CLI commands include `portStatsShow` and `portErrorShow` . Refer to the *Brocade Fabric OS Command Reference* for information and examples of these commands.

Retrieving a List of Logical E_Ports

This example uses a GET request to retrieve a list of logical E_Ports from logical switch 128.

Structure

```
GET base_URI/running/brocade-interface/logical-e-port
```

URI

```
GET https://10.10.10.10/rest/running/brocade-interface/logical-e-port
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the header.

```
        {
    "Response": {
        "logical-e-port": [
            {
                "name": 64,
                "fabric-id": 128,
                "operational-status": 2,
                "associated-physical-ports": {
                    "port": [
                        "42",
                        "1"
                    ]
                }
            }
        ]
    }
}
```

Getting Details on a Single Fibre Channel Port

This example uses a GET request to get the details of a single Fibre Channel port; in this example, port 7/1.

Structure

```
GET <base_URI>/running/brocade-interface/fibrechannel/name/fibre-channel-interface-name
```

URI

Notice that the slash character in the name resource is encoded as “%2f”.

```
GET https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/7%2f1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the header.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel>
        <name>7/1</name>
        <wwn>20:81:c4:f5:7c:4a:ac:6c</wwn>
        <port-type>7</port-type>
        <port-type-string>e-port</port-type-string>
        <speed>16000000000</speed>
        <max-speed>32000000000</max-speed>
        <user-friendly-name>slot7 port1</user-friendly-name>
        <operational-status>2</operational-status>
        <enabled-state>2</enabled-state>
        <is-enabled-state>true</is-enabled-state>
        <auto-negotiate>1</auto-negotiate>
        <isl-ready-mode-enabled>0</isl-ready-mode-enabled>
        <long-distance>0</long-distance>
        <trunk-port-enabled>1</trunk-port-enabled>
        <vc-link-init>0</vc-link-init>
        <measured-link-distance>lt 2</measured-link-distance>
        <pod-license-status>true</pod-license-status>
        <index>129</index>
        <default-index>129</default-index>
        <fcid>2851072</fcid>
        <fcid-hex>0x2b8100</fcid-hex>
        <physical-state>online</physical-state>
        <persistent-disable>0</persistent-disable>
        <e-port-disable>0</e-port-disable>
        <ex-port-enabled>0</ex-port-enabled>
        <npiv-enabled>1</npiv-enabled>
        <npiv-pp-limit>126</npiv-pp-limit>
        <qos-enabled>1</qos-enabled>
        <csctl-mode-enabled>0</csctl-mode-enabled>
        <port-autodisable-enabled>0</port-autodisable-enabled>
        <e-port-credit>0</e-port-credit>
        <d-port-enable>0</d-port-enable>
        <octet-speed-combo>1</octet-speed-combo>
        <encryption-enabled>0</encryption-enabled>
        <compression-configured>0</compression-configured>
        <compression-active>0</compression-active>
        <encryption-active>0</encryption-active>
        <mirror-port-enabled>0</mirror-port-enabled>
        <fec-enabled>1</fec-enabled>
        <fec-active>1</fec-active>
        <via-tts-fec-enabled>0</via-tts-fec-enabled>
        <port-health>healthy</port-health>
        <authentication-protocol>none</authentication-protocol>
        <port-peer-beacon-enabled>false</port-peer-beacon-enabled>
        <neighbor>
            <wwn>20:5c:00:27:f8:66:f3:81</wwn>
        </neighbor>
        <neighbor-node-wwn>10:00:00:27:f8:66:f3:81</neighbor-node-wwn>
```

```
        <neighbor-port-index>92</neighbor-port-index>
        <neighbor-switch-user-friendly-name>66-92_B6520_Odin</neighbor-switch-user-friendly-name>
        <le-domain>0</le-domain>
        <target-driven-zoning-enable>0</target-driven-zoning-enable>
        <npiv-flogi-logout-enabled>0</npiv-flogi-logout-enabled>
        <sim-port-enabled>0</sim-port-enabled>
        <f-port-buffers>0</f-port-buffers>
        <clean-address-enabled>false</clean-address-enabled>
        <reserved-buffers>20</reserved-buffers>
        <average-transmit-buffer-usage>0</average-transmit-buffer-usage>
        <average-transmit-frame-size>0</average-transmit-frame-size>
        <average-receive-buffer-usage>0</average-receive-buffer-usage>
        <average-receive-frame-size>0</average-receive-frame-size>
        <current-buffer-usage>34</current-buffer-usage>
        <recommended-buffers>34</recommended-buffers>
        <chip-instance>0</chip-instance>
        <chip-buffers-available>7572</chip-buffers-available>
        <fault-delay-enabled>0</fault-delay-enabled>
        <credit-recovery-enabled>1</credit-recovery-enabled>
        <credit-recovery-active>1</credit-recovery-active>
        <rscn-suppression-enabled>0</rscn-suppression-enabled>
        <los-tov-mode-enabled>0</los-tov-mode-enabled>
        <segmentation-reason/>
    </fibrechannel>
</Response>
```

Getting Details on a Single Fibre Channel Port Attribute

This example uses a GET request to determine if trunking is enabled on a port; in this example, port 3/4.

NOTE
You can get details for any port attribute in the fibrechannel-interface container.

Structure

```
GET <base_URI>/running/brocade-interface/fibrechannel/name/fibre-channel-interface-name/trunk-port-enabled
```

URI

Notice that the slash character in the name resource is encoded as “%2f”.

```
GET https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/3%2f4/trunk-port-enabled
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the header.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel>
        <name>3/4</name>
        <trunk-port-enabled>1</trunk-port-enabled>
```

```
    </fibrechannel>
</Response>
```

Configuring a Single Fibre Channel Port Attribute

This example uses a PATCH request to disable trunking on a port; in this example, port 3/4.

NOTE
You can configure any read-write port attribute in the fibrechannel container.

Structure

```
PATCH <base_URI>/running/brocade-interface/fibrechannel/name/fibre-channel-interface-name/trunk-port-enabled/0
```

URI

Notice that the slash character in the name resource is encoded as “%2f”.

```
PATCH https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/3%2f4/trunk-port-enabled/0
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty response body, and a “204 No Content” status appears in the header.

Setting the User-Friendly Name for a Fibre Channel Interface

This example uses a PATCH request to set the user-friendly name for Fibre Channel interface 10/2.

Structure

```
PATCH  <base_URI>/running/brocade-interface/fibrechannel/name/fibre-channel-interface-name/user-friendly-name/user-friendly-name
```

URI

Notice that the slash character in the name resource is encoded as “%2f”.

```
PATCH https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/10%2f2/user-friendly-name/Port_ten-two
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status appears in the header.

Setting a Fibre Channel Interface State

This example uses a PATCH request to disable Fibre Channel interface 10/2.

Structure

```
PATCH <base_URI>/running/brocade-interface/fibrechannel/name/fc_interface_name/is-enabled-state/enabled-state
```

URI

Notice that the slash character in the name resource is encoded as “%2f”.

```
PATCH https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/10%2f2/is-enabled-state/false
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body, and a “204 No Content” status appears in the header.

Getting Trunk Settings for a Physical Port

This example uses a GET request to get the trunk details of a physical port; in this example, port 0/88.

Structure

```
GET <base_URI>/running/brocade-interface/fibrechannel/name/fibre-channel-interface-name/trunk-enabled
```

URI

Notice that the slash character in the name resource is encoded as “%2f”.

```
GET https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/0%2f88/trunk-enabled
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the header.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel>
        <name>0/88</name>
        <trunk-port-enabled>1</trunk-port-enabled>
    </fibrechannel>
</Response>
```

brocade-interface/gigabitethernet

The Gigabit Ethernet parameters in this module allow you to retrieve or configure both Gigabit Ethernet interfaces and Gigabit Ethernet interface statistics.

For a list of changes to this module by release, refer to the individual module YANG file.

NOTE
The brocade-fibrechannel, brocade-extension-ip-interface, and brocade-gigabitethernet modules have been merged into the brocade-interface module in Fabric OS 8.2.1b or later. However, this section only covers brocade-gigabitethernet. For information about brocade-fibrechannel, refer to brocade-interface/fibrechannel. For information about brocade-extension-ip-interface, refer to brocade-interface/extension-ip-interface.

Module Tree

This is the tree view of the module from the brocade-gigabitethernet.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-interface
  +--rw brocade-interface
     +--rw gigabitethernet* [name] {fibrechannel:fibrechannel_extension_platform}?
     |  +--rw name                          brocade-interface-types:ge-interface-type
     |  x--rw enabled-state?                uint8
     |  +--rw enabled-state-v2?             boolean
     |  x--rw speed?                        brocade-interface-types:speed-type
     |  +--rw protocol-speed?               fibrechannel:protocol-speed-type
     |  +--rw fec?                          brocade-interface-types:fec-type
     |  +--ro mac-address?                  yang:mac-address
     |  +--ro operational-status?           uint16
     |  x--rw persistent-disable?           uint8
     |  +--rw persistent-disable-v2?        boolean
     |  +--rw protocol?                     extension:ge-protocol-type
     |  +--rw auto-negotiation-enabled?     boolean
     |  +--ro portchannel-name?             brocade-interface-types:portchannel-interface-name-type
     |  +--rw portchannel-member-timeout?   lag-port-timeout
     |  +--rw lldp-profile?                 string
     |  +--rw lldp-enabled-state?           boolean
     |  +--ro extension-vlans
     |     +--ro vlan*   uint32
     +--rw gigabitethernet-statistics* [name] {fibrechannel:fibrechannel_extension_platform}?
     |  +--rw name                  brocade-interface-types:ge-interface-type
     |  +--ro out-pkts?             yang:zero-based-counter64
     |  +--ro out-octets?           yang:zero-based-counter64
     |  +--ro out-unicast-pkts?     yang:zero-based-counter64
     |  +--ro out-multicast-pkts?   yang:zero-based-counter64
     |  +--ro out-broadcast-pkts?   yang:zero-based-counter64
     |  +--ro out-vlan-pkts?        yang:zero-based-counter64
     |  +--ro out-pause-pkts?       yang:zero-based-counter64
     |  +--ro in-pkts?              yang:zero-based-counter64
     |  +--ro in-octets?            yang:zero-based-counter64
     |  +--ro in-unicast-pkts?      yang:zero-based-counter64
     |  +--ro in-multicast-pkts?    yang:zero-based-counter64
     |  +--ro in-broadcast-pkts?    yang:zero-based-counter64
     |  +--ro in-vlan-pkts?         yang:zero-based-counter64
```

```
|  +--ro in-pause-pkts?        yang:zero-based-counter64
|  +--ro carrier-loss-error?   yang:zero-based-counter64
|  +--ro crc-error?            yang:zero-based-counter64
|  +--ro jabber-error?         yang:zero-based-counter64
|  x--rw reset-statistics?     uint8
|  +--ro time-generated?       fibrechannel:time-generated-type
```

Examples

This section provides a few examples for using the brocade-interface/gigabitethernet module.

Retrieving Gigabit Ethernet Port Statistics

This example uses a GET request to retrieve the Gigabit Ethernet port statistics for all Gigabit Ethernet ports on the switch.

Structure

```
GET <base_URI>/running/brocade-interface/gigabitethernet-statistics
```

URI

```
GET https://10.10.10.10/rest/running/brocade-interface/gigabitethernet-statistics
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
  <gigabitethernet-statistics>
    <name>4/0</name>
    <out-pkts>178269</out-pkts>
    <out-octets>23396402</out-octets>
    <out-unicast-pkts>176862</out-unicast-pkts>
    <out-multicast-pkts>0</out-multicast-pkts>
    <out-broadcast-pkts>1407</out-broadcast-pkts>
    <out-vlan-pkts>0</out-vlan-pkts>
    <out-pause-pkts>0</out-pause-pkts>
    <in-pkts>278070</in-pkts>
    <in-octets>30069936</in-octets>
    <in-unicast-pkts>267797</in-unicast-pkts>
    <in-multicast-pkts>0</in-multicast-pkts>
    <in-broadcast-pkts>10273</in-broadcast-pkts>
    <in-vlan-pkts>0</in-vlan-pkts>
    <in-pause-pkts>0</in-pause-pkts>
    <carrier-loss-error>3</carrier-loss-error>
    <crc-error>0</crc-error>
    <jabber-error>0</jabber-error>
    <reset-statistics>0</reset-statistics>
    <time-generated>1500431065</time-generated>
```

```
  </gigabitethernet-statistics>
...
  <gigabitethernet-statistics>
    <name>4/17</name>
    <out-pkts>0</out-pkts>
    <out-octets>0</out-octets>
    <out-unicast-pkts>0</out-unicast-pkts>
    <out-multicast-pkts>0</out-multicast-pkts>
    <out-broadcast-pkts>0</out-broadcast-pkts>
    <out-vlan-pkts>0</out-vlan-pkts>
    <out-pause-pkts>0</out-pause-pkts>
    <in-pkts>0</in-pkts>
    <in-octets>0</in-octets>
    <in-unicast-pkts>0</in-unicast-pkts>
    <in-multicast-pkts>0</in-multicast-pkts>
    <in-broadcast-pkts>0</in-broadcast-pkts>
    <in-vlan-pkts>0</in-vlan-pkts>
    <in-pause-pkts>0</in-pause-pkts>
    <carrier-loss-error>0</carrier-loss-error>
    <crc-error>0</crc-error>
    <jabber-error>0</jabber-error>
    <reset-statistics>0</reset-statistics>
    <time-generated>1500431065</time-generated>
  </gigabitethernet-statistics>
</Response>
```

Retrieving Information for a Single Gigabit Ethernet Port

This example uses a GET request to retrieve information about the Gigabit Ethernet port “0/17”.

Structure

```
GET <base_URI>/running/brocade-interface/gigabitethernet/name/<port>
```

URI

Notice that the slash character in the name resource is encoded as “%2f”.

```
GET https://10.10.10.10/rest/running/brocade-interface/gigabitethernet/name/0%2f17
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "gigabitethernet": {
            "name": "0/17",
            "enabled-state": 1,
            "enabled-state-v2": true,
            "speed": 100000000000,
            "protocol-speed": "100-ge",
            "mac-address": "c4:f5:7c:02:8b:1a",
```

```
            "operational-status": 2,
            "persistent-disable": 0,
            "persistent-disable-v2": false,
            "portchannel-name": "",
            "lldp-profile": "",
            "lldp-enabled-state": true,
            "extension-vlans": {},
            "protocol": "WAN",
            "fec": "cl91"
        }
    }
}
```


Resetting the Port Stats for a Single Gigabit Ethernet Port

This example uses a PATCH request to reset statistics for the Gigabit Ethernet port 4/2. The “name” key/value is included in the URI to indicate the correct port.

Structure

```
PATCH <base_URI>/running/brocade-interface/gigabitethernet/name/name/reset-statistics/1
```

Request Body

No request body is required.

Response Body

When the statistics are is successfully reset, the response contains an empty message body and a "204 No Content" status in the headers.

brocade-interface/portchannel

This module enables you to view and configure port channel interfaces on the switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the brocade-interface/portchannel module from the brocade-interface.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-interface
  +--rw brocade-interface
     +--rw portchannel* [name] {fibrechannel:portchannel_platform}?
     |  +--rw name                                brocade-interface-types:portchannel-interface-name-type
     |  +--rw key?                                uint32
     |  +--rw portchannel-type?                   portchannel-type
     |  x--ro speed?                              brocade-interface-types:portchannel-speed-type
     |  +--ro protocol-speed?                     fibrechannel:protocol-speed-type
     |  +--ro lacp-system-mac-address?            ietf:mac-address
     |  +--rw admin-state-enabled?                boolean
     |  +--rw auto-negotiation-enabled?           boolean
     |  +--ro operational-status-enabled?         boolean
     |  +--rw gigabit-ethernet-member-ports
     |  |  +--rw port*   brocade-interface-types:ge-interface-type
     |  +--ro primary-member-gigabit-ethernet-port?   brocade-interface-types:ge-interface-type
     +--ro portchannel-statistics* [name] {fibrechannel:portchannel_platform}?
     |  +--ro name                  brocade-interface-types:portchannel-interface-name-type
     |  +--ro out-pkts?             yang:zero-based-counter64
     |  +--ro out-octets?           yang:zero-based-counter64
     |  +--ro out-unicast-pkts?     yang:zero-based-counter64
     |  +--ro out-multicast-pkts?   yang:zero-based-counter64
     |  +--ro out-broadcast-pkts?   yang:zero-based-counter64
     |  +--ro out-vlan-pkts?        yang:zero-based-counter64
     |  +--ro out-pause-pkts?       yang:zero-based-counter64
     |  +--ro in-pkts?              yang:zero-based-counter64
     |  +--ro in-octets?            yang:zero-based-counter64
     |  +--ro in-unicast-pkts?      yang:zero-based-counter64
     |  +--ro in-multicast-pkts?    yang:zero-based-counter64
     |  +--ro in-broadcast-pkts?    yang:zero-based-counter64
     |  +--ro in-vlan-pkts?         yang:zero-based-counter64
     |  +--ro in-pause-pkts?        yang:zero-based-counter64
     |  +--ro carrier-loss-error?   yang:zero-based-counter64
     |  +--ro crc-error?            yang:zero-based-counter64
     |  +--ro jabber-error?         yang:zero-based-counter64
```

Examples

This section provides basic examples for using the brocade-interface/portchannel module.

Displaying Port Channel Details

This example uses a GET request to display the port channel details.

Structure

```
GET base_URI/brocade-interface/portchannel/name/port_channel_name
```

URI

```
GET https://10.10.10.10/rest/running/brocade-interface/portchannel/name/port_channel_name
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <portchannel>
        <name>rest-test4</name>
        <lacp-system-mac-address>88:94:71:52:c6:8e</lacp-system-mac-address>
        <portchannel-type>dynamic</portchannel-type>
        <speed>1000000000</speed>
        <admin-state-enabled>true</admin-state-enabled>
        <auto-negotiation-enabled>true</auto-negotiation-enabled>
        <operational-status-enabled>false</operational-status-enabled>
        <primary-member-gigabit-ethernet-port/>
        <gigabit-ethernet-member-ports>
            <port>0/0</port>
            <port>0/1</port>
        </gigabit-ethernet-member-ports>
        <key>999</key>
    </portchannel>
    <portchannel>
        <name>dynamic</name>
        <lacp-system-mac-address>c4:f5:7c:8d:b2:ce</lacp-system-mac-address>
        <portchannel-type>dynamic</portchannel-type>
        <speed>10000000000</speed>
        <admin-state-enabled>true</admin-state-enabled>
        <operational-status-enabled>true</operational-status-enabled>
        <primary-member-gigabit-ethernet-port>5/2</primary-member-gigabit-ethernet-port>
        <gigabit-ethernet-member-ports>
            <port>5/2</port>
            <port>5/3</port>
        </gigabit-ethernet-member-ports>
        <key>2</key>
    </portchannel>
</Response>
```

Configuring a Port Channel

This example uses a POST request to create a port channel 'dlag101'.

Structure

```
POST base_URI/brocade-interface/portchannel/name/port_channel_name
```

URI

```
POST https://10.10.10.10/rest/brocade-interface/portchannel/name/dlag101
```

Request Body

```
<portchannel>
        <portchannel-type>dynamic</portchannel-type>
        <admin-state-enabled>true</admin-state-enabled>
        <gigabit-ethernet-member-ports>
            <port>5/4</port>
            <port>5/5</port>
        </gigabit-ethernet-member-ports>
        <key>6</key>
</portchannel>
```

Response Body

When the operation is successful, there is no response, and a “201 Created” status appears in the headers.

brocade-license

This module provides a detailed view of the licenses installed on the switch.

For a list of changes to this module by release, refer to the individual module YANG file.

Module tree

This is the tree view of the module from the `brocade-license.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-license
  +--ro brocade-license
     +--ro license* [name]
     |  +--ro name                        string
     |  +--ro features
     |  |  +--ro feature*   string
     |  +--ro capacity?                   uint32
     |  +--ro consumed?                   uint32
     |  +--ro configured-blade-slots
     |  |  +--ro configured-blade-slot*   uint32
     |  +--ro expiration-date?            string
     |  +--ro generation-date?            string
     |  +--ro active-firmware-version?    fru:firmware-version-type
     |  +--ro target-firmware-version?    fru:firmware-version-type
     |  +--ro license-format?             enumeration
     +--ro end-user-license-agreement
     |  +--ro text?   string
     +--ro ports-on-demand-license-info
        +--ro sfp-port-count?                                    yang:gauge32
        +--ro qsfp-port-count?                                   yang:gauge32
        +--ro sfp-dd-count?                                      yang:gauge32
        +--ro port-on-demand-license?                            boolean
        +--ro qflex-port-on-demand-license?                      boolean
        +--ro sfp-dd-port-on-demand-license?                     boolean
        +--ro provisioned-sfp-port-count?                        yang:gauge32
        +--ro sfp-port-base-switch-allowance-count?              yang:gauge32
        +--ro sfp-port-license-allowance-count?                  yang:gauge32
        +--ro sfp-port-assigned-count?                           yang:gauge32
        +--ro sfp-port-base-switch-assigned-count?               yang:gauge32
        +--ro licensed-sfp-port-count?                           yang:gauge32
        +--ro sfp-port-license-reservation-available-count?      yang:gauge32
        +--ro sfp-port-base-switch-members
        |  +--ro port*   fibrechannel:port-number-type
        +--ro licensed-sfp-port-members
        |  +--ro port*   fibrechannel:port-number-type
        +--ro unassigned-sfp-port-members
        |  +--ro port*   fibrechannel:port-number-type
        +--ro licensed-offline-sfp-port-members
        |  +--ro port*   fibrechannel:port-number-type
        +--ro provisioned-qsfp-port-count?                       yang:gauge32
        +--ro qsfp-base-switch-allowance-count?                  yang:gauge32
        +--ro qsfp-license-allowance-count?                      yang:gauge32
        +--ro qsfp-port-assigned-count?                          yang:gauge32
```

```
   +--ro qsfp-port-base-switch-assigned-count?             yang:gauge32
   +--ro licensed-qsfp-port-count?                         yang:gauge32
   +--ro qsfp-port-license-reservation-available-count?    yang:gauge32
   +--ro qsfp-port-base-switch-members
   |  +--ro port*   fibrechannel:port-number-type
   +--ro licensed-qsfp-port-members
   |  +--ro port*   fibrechannel:port-number-type
   +--ro unassigned-qsfp-port-members
   |  +--ro port*   fibrechannel:port-number-type
   +--ro licensed-offline-qsfp-port-members
   |  +--ro port*   fibrechannel:port-number-type
   +--ro provisioned-sfp-dd-port-count?                    yang:gauge32
   +--ro sfp-dd-base-switch-allowance-count?               yang:gauge32
   +--ro sfp-dd-license-allowance-count?                   yang:gauge32
   +--ro sfp-dd-port-assigned-count?                       yang:gauge32
   +--ro sfp-dd-port-base-switch-assigned-count?           yang:gauge32
   +--ro licensed-sfp-dd-port-count?                       yang:gauge32
   +--ro sfp-dd-port-license-reservation-available-count?  yang:gauge32
   +--ro sfp-dd-port-base-switch-members
   |  +--ro port*   fibrechannel:port-number-type
   +--ro licensed-sfp-dd-port-members
   |  +--ro port*   fibrechannel:port-number-type
   +--ro unassigned-sfp-dd-port-members
   |  +--ro port*   fibrechannel:port-number-type
   +--ro licensed-offline-sfp-dd-port-members
      +--ro port*   fibrechannel:port-number-type
```

Examples

This section provides basic examples for using the brocade-license module.

Viewing the License Data

The following example uses the GET request to display a detailed view of the licenses installed on the switch.

Structure

```
GET <base_URI>/running/brocade-license/license
```

URI

```
GET https://10.10.10.10/rest/running/brocade-license/license
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
      <license>
        <name>QYRKQTWJNfCZHCf3NNfBgJLWYWCDrrLSB7BZN</name>
        <features>
            <feature>Extended Fabric</feature>
```

```
            <feature>Trunking</feature>
            <feature>FICON_CUP</feature>
            <feature>Fabric Vision and IO Insight</feature>
        </features>
        <license-format>key-based-license</license-format>
    </license>
    <license>
        <name>rCTRGSBSB9BBPZEtAYB4g7ZTZ3C974PQANXNCZFABHDB</name>
        <features>
            <feature>Q-Flex Ports on Demand</feature>
        </features>
        <capacity>32</capacity>
        <license-format>key-based-license</license-format>
    </license>
    <license>
        <name>AgP4QaZ3J7LNSKRDTfrg9GgLQD3A4DSfLHYf4THAfaAA</name>
        <features>
            <feature>Ports on Demand</feature>
        </features>
        <capacity>48</capacity>
        <license-format>key-based-license</license-format>
    </license>
    <license>
        <name>PPRgJXLGGXDQ9DHNrRK9HTCGS7HSAHDtMWSD4SGAZKFA</name>
        <features>
            <feature>Integrated Routing Ports on Demand</feature>
        </features>
        <capacity>128</capacity>
        <license-format>key-based-license</license-format>
    </license>
</Response>
```

Viewing the Port License Assignments

This example uses a GET request to display the port license assignments..

Structure

```
GET base_URI/running/brocade-license/ports-on-demand-license-info
```

URI

```
GET https://10.10.10.10/rest/running/brocade-license/ports-on-demand-license-info
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <ports-on-demand-license-info>
        <sfp-port-count>96</sfp-port-count>
        <qsfp-port-count>32</qsfp-port-count>
```

```
<sfp-dd-count>0</sfp-dd-count>
<port-on-demand-license>true</port-on-demand-license>
<qflex-port-on-demand-license>true</qflex-port-on-demand-license>
<sfp-dd-port-on-demand-license>false</sfp-dd-port-on-demand-license>
<provisioned-sfp-port-count>96</provisioned-sfp-port-count>
<sfp-port-base-switch-allowance-count>48</sfp-port-base-switch-allowance-count>
<sfp-port-license-allowance-count>48</sfp-port-license-allowance-count>
<sfp-port-assigned-count>96</sfp-port-assigned-count>
<sfp-port-base-switch-assigned-count>48</sfp-port-base-switch-assigned-count>
<licensed-sfp-port-count>48</licensed-sfp-port-count>
<sfp-port-license-reservation-available-count>0</sfp-port-license-reservation-available-count>
<sfp-port-base-switch-members>
    <sfp-port-base-switch-members>0</sfp-port-base-switch-members>
    <sfp-port-base-switch-members>1</sfp-port-base-switch-members>
    <sfp-port-base-switch-members>2</sfp-port-base-switch-members>
    <sfp-port-base-switch-members>3</sfp-port-base-switch-members>
    ...
    <sfp-port-base-switch-members>93</sfp-port-base-switch-members>
    <sfp-port-base-switch-members>94</sfp-port-base-switch-members>
    <sfp-port-base-switch-members>95</sfp-port-base-switch-members>
</sfp-port-base-switch-members>
<licensed-sfp-port-members>
    <licensed-sfp-port-members>10</licensed-sfp-port-members>
    <licensed-sfp-port-members>11</licensed-sfp-port-members>
    <licensed-sfp-port-members>12</licensed-sfp-port-members>
    ...
    <licensed-sfp-port-members>58</licensed-sfp-port-members>
    <licensed-sfp-port-members>59</licensed-sfp-port-members>
    <licensed-sfp-port-members>61</licensed-sfp-port-members>
    <licensed-sfp-port-members>63</licensed-sfp-port-members>
</licensed-sfp-port-members>
<unassigned-sfp-port-members/>
<licensed-offline-sfp-port-members>
    <licensed-offline-sfp-port-members>0</licensed-offline-sfp-port-members>
    <licensed-offline-sfp-port-members>1</licensed-offline-sfp-port-members>
    <licensed-offline-sfp-port-members>2</licensed-offline-sfp-port-members>
    ...
    <licensed-offline-sfp-port-members>93</licensed-offline-sfp-port-members>
    <licensed-offline-sfp-port-members>94</licensed-offline-sfp-port-members>
    <licensed-offline-sfp-port-members>95</licensed-offline-sfp-port-members>
</licensed-offline-sfp-port-members>
<provisioned-qsfp-port-count>32</provisioned-qsfp-port-count>
<qsfp-base-switch-allowance-count>0</qsfp-base-switch-allowance-count>
<qsfp-license-allowance-count>32</qsfp-license-allowance-count>
<qsfp-port-assigned-count>32</qsfp-port-assigned-count>
<qsfp-port-base-switch-assigned-count>0</qsfp-port-base-switch-assigned-count>
<licensed-qsfp-port-count>32</licensed-qsfp-port-count>
<qsfp-port-license-reservation-available-count>0</qsfp-port-license-reservation-available-count>
<qsfp-port-base-switch-members/>
<licensed-qsfp-port-members>
    <licensed-qsfp-port-members>96</licensed-qsfp-port-members>
    <licensed-qsfp-port-members>97</licensed-qsfp-port-members>
    <licensed-qsfp-port-members>98</licensed-qsfp-port-members>
```

```
            ...
            <licensed-qsfp-port-members>125</licensed-qsfp-port-members>
            <licensed-qsfp-port-members>126</licensed-qsfp-port-members>
            <licensed-qsfp-port-members>127</licensed-qsfp-port-members>
        </licensed-qsfp-port-members>
        <unassigned-qsfp-port-members/>
        <licensed-offline-qsfp-port-members>
            <licensed-offline-qsfp-port-members>96</licensed-offline-qsfp-port-members>
            <licensed-offline-qsfp-port-members>97</licensed-offline-qsfp-port-members>
            <licensed-offline-qsfp-port-members>98</licensed-offline-qsfp-port-members>
            ...
            <licensed-offline-qsfp-port-members>125</licensed-offline-qsfp-port-members>
            <licensed-offline-qsfp-port-members>126</licensed-offline-qsfp-port-members>
            <licensed-offline-qsfp-port-members>127</licensed-offline-qsfp-port-members>
        </licensed-offline-qsfp-port-members>
        <provisioned-sfp-dd-port-count>0</provisioned-sfp-dd-port-count>
        <sfp-dd-base-switch-allowance-count>0</sfp-dd-base-switch-allowance-count>
        <sfp-dd-license-allowance-count>0</sfp-dd-license-allowance-count>
        <sfp-dd-port-assigned-count>0</sfp-dd-port-assigned-count>
        <sfp-dd-port-base-switch-assigned-count>0</sfp-dd-port-base-switch-assigned-count>
        <licensed-sfp-dd-port-count>0</licensed-sfp-dd-port-count>
        <sfp-dd-port-license-reservation-available-count>0</sfp-dd-port-license-reservation-available-count>
        <sfp-dd-port-base-switch-members/>
        <licensed-sfp-dd-port-members/>
        <unassigned-sfp-dd-port-members/>
        <licensed-offline-sfp-dd-port-members/>
    </ports-on-demand-license-info>
</Response>
```

Viewing the Brocade End User License Agreement

This example uses a GET request to display the Brocade End User License Agreement.

Structure

```
GET base_URI/running/brocade-license/end-user-license-agreement
```

URI

```
GET https://10.10.10.10/rest/running/brocade-license/end-user-license-agreement
```

Request Body

No request body is required.

Response Body

```
{
 "Response": {
  "end-user-license-agreement": {
   "text": "*************************************************************************
 \n* BROCADE FABRIC OPERATING SYSTEM AND FEATURE LICENSES AND LICENSE KEYS *
\n*************************************************************************\n\n\nEND USER LICENSE AGREEMENT
\n==========================\n\nTHIS END USER LICENSE AGREEMENT (\"Agreement\") GOVERNS THE DOWNLOAD,
 INSTALLATION,\nUSE, POSTING, DISTRIBUTING AND OTHERWISE MAKING AVAILABLE OF BROCADE'S FABRIC \nOPERATING
 SYSTEM (\"FOS\") SOFTWARE AND/ OR USE OF BROCADE FEATURE LICENSES AND \nLICENSE KEYS THAT ACTIVATE FOS OR
```

FUNCTIONALITY WITHIN FOS, AND ACCOMPANYING \nDOCUMENTATION (collectively the \"Software\"). BY DOWNLOADING, INSTALLING, USING, \nPOSTING, DISTRIBUTING OR OTHERWISE MAKING AVAILABLE THE SOFTWARE, OR BY \nPURCHASING, CONVERTING A TRANSACTION KEY INTO A LICENSE KEY, OR INSTALLING A \nLICENSE OR LICENSE KEY, YOU ARE AGREEING TO BE BOUND ON AN ONGOING BASIS BY THE \nTERMS AND CONDITIONS HEREIN, WHICH MAY BE UPDATED BY BROCADE FROM TIME TO TIME. \nIF AT ANY TIME YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, PROMPTLY \nSTOP USE OF THE SOFTWARE AND DESTROY ALL COPIES OF THE SOFTWARE IN YOUR \nPOSSESSION OR CONTROL, AND CERTIFY IN WRITING TO BROCADE SUCH CESSATION OF USE \nAND DESTRUCTION.\n\nSingle User License\n==========\nSubject to the terms and conditions of this Agreement and payment of the applicable\nlicense fees, Brocade Communications Systems LLC (\"Brocade\") and its suppliers grant\nto you (\"End User\") a non-exclusive, non-transferable, non-assignable, non-sub licensable\nlicense to use the Software in object code form (in the case of FOS) solely for the\npurpose of operating Brocade storage area networking switches in accordance with the\nsection entitled \"Restrictions\" below. End User may make such backup copies of the\nSoftware as may be necessary for End User's lawful use, provided End User affixes to\nsuch copies all trademark, copyright, patent, and notices of other proprietary rights\nthat appear on the original.\n\nFOS Licenses and License Keys\n==========\nFOS Licenses are used to activate FOS features or functionality in Brocade switches.\nFOS Licenses shall only be purchased from a Brocade authorized distributor or official\nOEM partner, and FOS License Keys shall only be obtained through Brocade's License Portal\nor an authorized OEM License Portal and shall only be used on Brocade switches purchased\nthrough authorized channels. FOS License Keys obtained from any other sources are not\npermitted to be used on any Brocade hardware products. FOS Licenses and License Keys\nshall not be resold, assigned, sub-licensed or otherwise transferred to a third party.\n\nOpen Source Software\n==========\nCertain components of the Software may incorporate \"open source\" software. Open source\nsoftware is subject to the applicable open source license (e.g., the GNU General Public\nLicense) and is not subject to this Agreement. To obtain a copy of the source code and\napplicable licensing terms for the open source software used by Brocade, please download\nthe open source disclosure documents in the Broadcom Customer Support Portal (CSP). If\nyou do not have a CSP account or are unable to login, please contact your support\nprovider for this information. \n\nRestrictions\n==========\nFOS is designed for use with Brocade hardware products sold through authorized sales\nchannels, and with hardware platforms that are currently supported by Brocade. End User\nacknowledges that use of the Software on hardware products that do not meet the above\ncriteria, or which is inconsistent with the restrictions below, presents risks to the\nsecurity and quality of Products and to the Software, for which Brocade disclaims all\nliability. Except as otherwise expressly provided in this Agreement or expressly\nauthorized by Brocade, End User shall have no right, and End User specifically agrees\nnot to, and not to permit third parties to: (i) modify, adapt, or create derivative\nworks based upon the Software; (ii) copy, in whole or in part, decompile, translate,\nreverse engineer, disassemble or otherwise reduce the Software to humanreadable form;\n(iii) install the Software on any products other than Brocade storage area networking\nswitches purchased by End User through authorized channels for its own use or for \nresale expressly authorized by Brocade; (iv) install or use the Software on products\nin excess of the maximum number of Brocade storage area networking switches for which\nEnd User is licensed; (v) install the Software on any Brocade storage area networking\nswitches that have reached End of Support as defined by Brocade, or refurbished devices,\nexcept with Brocade's prior written consent, (vi) use the Software on any devices not\ndesignated by Brocade for use with the Software (as notified through software release\nnotes and other notices), (vii) download, or make available for use or download, any\nversions of the Software that have reached End of Support, as declared by Brocade, or\n(viii) sub-license or permit third parties to post or otherwise make available, or\nuse the Software to provide support for Brocade products unless expressly authorized\nby Brocade.\n\nUpgrades and Additional Copies\n==========\nFor purposes of this Agreement, \"Software\" shall also include any upgrades, updates,\nbug fixes or modified versions (\"Upgrades\") provided to End User by Brocade or an\nauthorized distributor and for which End User had paid the applicable license fees,\nand any backup copies of the Software. Notwithstanding the foregoing, End User\nacknowledges and agrees that Brocade and its resellers and distributors shall have\nno obligation to provide any Upgrades. If Upgrades or additional copies of the\nSoftware are provided, End User acknowledges and agrees that it has no license or\nright to use such additional copies or Upgrades unless End User, at the time of\nacquiring such copy or Upgrade, already holds a valid license to the original\nSoftware for the applicable number of copies.\n\nNo Support

\n==========\nBrocade may, but is not required to, provide technical support for the Software.\n\nSecurity Notice\n==========\nSoftware may include security features that End User can use, along with other security\ntools, to implement increased security in End User's storage area network (\"SAN\").\nHowever, use of the security features of the Software does not guarantee the security\nof End User's SAN or overall network. There are numerous factors that affect the security\nof a SAN, including, without limitation, correct installation and setup of the security\nfeatures of Software and all related requirements, correctly configured security\npolicies, selection of hardware and software (including network security tools), \ncorrect installation, configuration, and maintenance of the hardware and software,\nthe interoperability of the various components of End User's SAN and End User's network,\nand a physically and electronically secure operating environment. In addition, Software\nmay utilize digital certificates in connection with its access control features.\nAlthough digital certificates are a useful authentication security measure that improves\noverall security, they do not guarantee authenticity or security. In designing the\nsecurity of End User's SAN, it is End User's responsibility to evaluate all of these\nfactors to ensure End User's SAN meets End User's security requirements. End User's\nexperience may vary based on these and other factors. End User's use of Software,\nincluding the digital certificates, is subject to and governed by the terms of this\nAgreement and to End User's compliance with the policies and procedures for the use\nof the security features of Software and digital certificates, which may be made\navailable to End User by Brocade from time to time. If Brocade becomes aware of a breach\nof the security of its digital certificate infrastructure, Brocade reserves the right\nto re-issue digital certificates. In that event, End User will be required to submit\nnew certificate signing requests and install reissued certificates across End User's\nSAN. End User should plan for any network disruption that this may cause. END USER\nACKNOWLEDGES THAT END USER HAS ACCESS TO SUFFICIENT INFORMATION TO ENSURE THAT END USER\nCAN MAKE AN INFORMED DECISION AS TO THE EXTENT TO WHICH END USER CHOOSES TO RELY ON\nDIGITAL CERTIFICATES AND OTHER SECURITY FEATURES IN SOFTWARE (\"SECURITY\").\n\nDisclaimer of Warranty\n==========\nTO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, BROCADE PROVIDES THE SOFTWARE\n\"AS IS\" WITHOUT WARRANTY OF ANY KIND. BROCADE DOES NOT WARRANT THAT THE SOFTWARE\nIS ERROR FREE OR THAT END USER WILL BE ABLE TO OPERATE THE SOFTWARE WITHOUT\nPROBLEMS OR INTERRUPTIONS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW,\nBROCADE AND ITS LICENSORS, SUPPLIERS, DISTRIBUTORS AND RESELLERS DISCLAIM ALL\nEXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT\nLIMITATION, ANY IMPLIED WARRANTY OR CONDITION OF MERCHANTABILITY, FITNESS FOR A\nPARTICULAR PURPOSE, NONINFRINGEMENT, AND SATISFACTORY QUALITY OR ARISING FROM A\nCOURSE OF DEALING, USAGE, OR TRADE PRACTICE.\n\nLimitation of Liability\n==========\nIN NO EVENT WILL BROCADE OR ITS LICENSORS, SUPPLIERS, RESELLERS OR DISTRIBUTORS\nBE LIABLE FOR ANY LOST REVENUE, PROFIT, OR DATA, OR FOR SPECIAL, INDIRECT,\nCONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES HOWEVER CAUSED AND REGARDLESS\nOF THE THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE\nSOFTWARE EVEN IF BROCADE OR ITS LICENSORS, SUPPLIERS, RESELLERS OR DISTRIBUTORS\nHAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL BROCADE'S\nOR ITS LICENSORS', SUPPLIERS', RESELLERS' OR DISTRIBUTORS' TOTAL LIABILITY TO\nEND USER, WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, EXCEED\nTHE PRICE PAID BY END USER. THE FOREGOING LIMITATIONS SHALL APPLY EVEN IF THE\nABOVE-STATED WARRANTY FAILS OF ITS ESSENTIAL PURPOSE. BECAUSE SOME STATES OR \nJURISDICTIONS DO NOT ALLOW LIMITATION OR EXCLUSION OF CONSEQUENTIAL OR INCIDENTAL\nDAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO END USER.\n\nProprietary Rights\n==========\nThe Software, including all intellectual property rights therein, is and will\nremain the sole and exclusive property of Brocade or its licensors.\n\nTerm and Termination\n==========\nThis Agreement is effective until terminated. End User's license rights under this\nAgreement will terminate immediately without notice from Brocade if End User fails\nto comply with any provision of this Agreement. Upon termination, End User must stop\nusing the Software and destroy all copies of Software and the corresponding license\nkeys in its possession or control. \n\nCompliance with Law\n==========\nEach party agrees to comply with all applicable laws, rules and regulations in\nconnection with its activities under this Agreement. Without limiting the foregoing, \nEnd User acknowledges and agrees that the Software, including technical data, is \nsubject to United States export control laws, including the United States Export\nAdministration Act and its associated regulations, and may be subject to export or\nimport regulations in other countries. End User agrees to comply strictly with all\nsuch regulations and acknowledges that End User has the responsibility to obtain \nlicenses to export, re-export, or import the Software.\n\nRestricted Rights\n==========\nThe Software under this Agreement is commercial computer software as that term is\ndescribed in 48 C.F.R. 252.227-7014(a)(1).

```
 If acquired by or on behalf of a civilian\nagency, the U.S. Government acquires this commercial computer
 software and/or\ncommercial computer software documentation subject to the terms of this Agreement\nas
 specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data)\nof the Federal Acquisition
 Regulations (\"FAR\") and its successors. If acquired by\nor on behalf of any agency within the Department of
 Defense (\"DOD\"), the\nU.S. Government acquires this commercial computer software and/or commercial computer
\nsoftware documentation subject to the terms of this Agreement as specified in\n48 C.F.R. 227.7202-3 of the
 DOD FAR Supplement (\"DFAR\") and its successors.\n\nGeneral\n=======\nThis Agreement will bind and inure to
 the benefit of each party's successors and\nassigns, provided that End User may not assign or transfer this
 Agreement, in whole\nor in part, without Brocade's written consent. This Agreement shall be governed by\nand
 construed in accordance with the laws of the State of California,\nUnited States of America, as if performed
 wholly within the state and without giving\neffect to the principles of conflict of law. No failure of either
 party to exercise\nor enforce any of its rights under this Agreement will act as a waiver of such\nrights.
 If any portion hereof is found to be void or unenforceable, the remaining\nprovisions of this Agreement
 shall remain in full force and effect. This Agreement\nis the complete and exclusive agreement between the
 parties with respect to the\nsubject matter hereof, superseding and replacing any and all prior agreements,
\ncommunications, and understandings (both written and oral) regarding such subject\nmatter.\n\nEffective
 October 1, 2019\n"
  }
 }
}
```

brocade-lldp

This module enables you to configure and monitor LLDP.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-lldp.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-lldp
  +--rw brocade-lldp
     +--rw lldp-global
     |  +--rw multiplier?           uint32
     |  +--rw tx-interval?          uint32
     |  +--rw system-name?          string
     |  +--rw system-description?   string
     |  +--rw enabled-state?        boolean
     |  +--ro mandatory-tlvs
     |  |  +--ro tlv*   lldp-mandatory-tlv-name-type
     |  +--rw optional-tlvs
     |     +--rw tlv*   lldp-optional-tlv-name-type
     +--rw lldp-profile* [name] {fibrechannel:fibrechannel_chassis_platform or
  fibrechannel:fibrechannel_extension_platform}?
     |  +--rw name            string
     |  +--rw multiplier?     int32
     |  +--rw tx-interval?    int32
     |  +--rw enabled-tlvs
     |     +--rw tlv*   lldp-optional-tlv-name-type
     +--ro lldp-neighbor* [slot-port] {fibrechannel:fibrechannel_chassis_platform or
  fibrechannel:fibrechannel_extension_platform}?
     |  +--ro slot-port                fibrechannel:slot-port-name-type
     |  +--ro dead-interval?           uint32
     |  +--ro remaining-life?          uint32
     |  +--ro remote-interface-name?   string
     |  +--ro chassis-id?              string
     |  +--ro system-name?             string
     +--ro lldp-statistics* [slot-port] {fibrechannel:fibrechannel_chassis_platform or
  fibrechannel:fibrechannel_extension_platform}?
        +--ro slot-port           fibrechannel:slot-port-name-type
        +--ro out-frames?         ietf:zero-based-counter64
        +--ro in-frames?          ietf:zero-based-counter64
        +--ro frames-aged-out?    ietf:zero-based-counter64
        +--ro frames-discarded?   ietf:zero-based-counter64
        +--ro error-frames?       ietf:zero-based-counter64
        +--ro tlv-discarded?      ietf:zero-based-counter64
        +--ro tlv-unrecognized?   ietf:zero-based-counter64
```

Examples

This section provides basic examples for using the brocade-lldp module.

Viewing Global LLDP Information for a Switch

This example uses a GET request to view the LLDP global information for the switch.

Structure

GET *base_URI*/running/brocade-lldp/lldp-global

URI

```
GET  https://10.10.10.10/rest/running/brocade-lldp/lldp-global
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <lldp-global>
        <multiplier>4</multiplier>
        <tx-interval>30</tx-interval>
        <system-name>Switch_Test</system-name>
        <system-description>Brocade_X7_4_Fabric_OS_Version_9_0_0</system-description>
        <enabled-state>true</enabled-state>
        <mandatory-tlvs>
            <tlv>chassis-id</tlv>
            <tlv>port-id</tlv>
            <tlv>time-to-live</tlv>
        </mandatory-tlvs>
        <optional-tlvs>
            <tlv>port-desc</tlv>
            <tlv>sys-name</tlv>
            <tlv>dcbx</tlv>
            <tlv>fcoe-app</tlv>
            <tlv>fcoe-lls</tlv>
        </optional-tlvs>
    </lldp-global>
</Response>
```

Configuring an LLDP Profile

This example uses a POST request to create an LLDP profile.

Structure

POST *base_URI/*running/brocade-lldp/lldp-profile/name/[name]

URI

```
POST https://10.10.10.10/rest/running/brocade-lldp/lldp-profile/name/test_profile
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Monitoring an LLDP Neighbor

This example uses a GET request to monitor the LLDP neighbor details for a specified port.

Structure

GET *base_URI*running/brocade-lldp/lldp-neighbor/slot-port/<slot%port>

URI

```
GET https://10.10.10.10/rest/running/brocade-lldp/lldp-neighbor/slot-port/8%2f2
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <lldp-neighbor>
        <slot-port>8/2</slot-port>
        <dead-interval>120</dead-interval>
        <remaining-life>102</remaining-life>
        <remote-interface-name>ge2</remote-interface-name>
        <chassis-id>c4f5.7cdc.c0fa</chassis-id>
        <system-name>B7840_073_017</system-name>
    </lldp-neighbor>
</Response>
```

Viewing LLDP Statistics

This example uses a GET request to view the LLDP statistics for a specified port.

Structure

GET *base_URI*running/brocade-lldp/lldp-statistics/slot-port/<slot%port>

URI

```
GET https://10.10.10.10/rest/running/brocade-lldp/lldp-statistics/slot-port/4%2f1
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <lldp-statistics>
```

```
        <slot-port>4/1</slot-port>
        <out-frames>0</out-frames>
        <frames-aged-out>0</frames-aged-out>
        <frames-discarded>0</frames-discarded>
        <error-frames>0</error-frames>
        <in-frames>0</in-frames>
        <tlv-discarded>0</tlv-discarded>
        <tlv-unrecognized>0</tlv-unrecognized>
    </lldp-statistics>
</Response>
```

brocade-logging

This module provides a detailed view of audit and RASlog message configuration.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-logging.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-logging
  +--rw brocade-logging
     +--rw audit
     |  +--rw audit-enabled?       boolean
     |  +--rw severity-level?      brocade-logging-types:severity-level-type
     |  +--rw filter-class-list
     |     +--rw filter-class*   brocade-logging-types:class-type
     +--rw syslog-server* [server]
     |  +--rw server         inet:host
     |  +--rw port?          inet:port-number
     |  +--rw secure-mode?   boolean
     +--rw raslog* [message-id]
     |  +--rw message-id          brocade-logging-types:message-id-type
     |  +--rw message-enabled?    boolean
     |  +--ro message-flooded?    boolean
     |  +--rw syslog-enabled?     boolean
     |  o--ro message-text?       brocade-logging-types:ascii-text-type
     |  +--rw current-severity?   union
     |  +--ro default-severity?   brocade-logging-types:severity-level-type
     +--rw raslog-module* [module-id]
     |  +--rw module-id      brocade-logging-types:module-id-type
     |  +--rw log-enabled?   boolean
     +--rw log-quiet-control* [log-type]
     |  +--rw log-type         string
     |  +--rw quiet-enabled    boolean
     |  +--rw start-time?      brocade-logging-types:time-24hr-type
     |  +--rw end-time?        brocade-logging-types:time-24hr-type
     |  +--rw days-of-week
     |     +--rw day*   string
     +--rw log-setting
     |  +--rw syslog-facility-level?      string
     |  +--rw syslog-source-ip-chassis?   boolean {fibrechannel:fibrechannel_chassis_platform}?
     |  +--rw keep-alive-period?          uint16
     |  +--rw clear-log?                  enumeration
     +--rw supportftp
     |  +--rw host?                          union
     |  +--rw user-name?                     string
     |  +--rw password?                      union
     |  +--rw remote-directory?              string
     |  +--rw auto-enabled?                  boolean
     |  +--rw protocol?                      union
     |  +--rw port?                          uint32
     |  +--rw connectivity-check-interval?   uint32
```

```
      +--ro audit-log* []
      |  +--ro sequence-number?             uint32
      |  +--ro time-stamp?                  yang:date-and-time
      |  +--ro message-id?                  brocade-logging-types:message-id-type
      |  +--ro switch-user-friendly-name?   string
      |  +--ro message-text?                string
      |  +--ro severity-level?              brocade-logging-types:severity-level-type
      |  +--ro event-class?                 brocade-logging-types:event-class-type
      |  +--ro ip-address?                  union
      |  +--ro user-name?                   union
      |  +--ro role?                        brocade-security-type:user-config-role-type
      |  +--ro interface?                   fibrechannel:interface-type
      |  +--ro application-name?            string
      |  +--ro fabric-id?                   uint32
      |  +--ro event-info?                  string
      |  +--ro application-user-name?       string
      +--ro error-log* []
      |  +--ro sequence-number?             uint32
      |  +--ro time-stamp?                  yang:date-and-time
      |  +--ro message-id?                  brocade-logging-types:message-id-type
      |  +--ro fabric-id?                   uint32
      |  +--ro slot-id?                     uint32
      |  +--ro severity-level?              brocade-logging-types:severity-level-type
      |  +--ro ffdc-generated-event?        boolean
      |  +--ro switch-user-friendly-name?   string
      |  +--ro message-text?                string
      |  +--ro event-info?                  string
      +--ro management-session-login-information* []
         +--ro timestamp?            yang:date-and-time
         +--ro host?                 inet:host
         +--ro user-name?            string
         +--ro application-name?     string
         +--ro session-id?           string
         +--ro vf-id?                fibrechannel:fabric-id-type
  {fibrechannel:fibrechannel_virtual_fabric_platform}?
         +--ro auth-token-present?   boolean
         +--ro protocol?             enumeration
         +--ro active?               boolean
```

Examples

This section provides examples for the brocade-logging module.

Viewing the Quiet Time Configuration

The following example uses the GET request to display the current quiet time configuration.

Structure

GET *<base_URI>*/running/brocade-logging/log-quiet-control

URI Request

```
GET https://10.10.10.10/rest/running/brocade-logging/log-quiet-control
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<Response>
    <log-quiet-control>
        <log-type>audit</log-type>
        <quiet-enabled>false</quiet-enabled>
        <days-of-week/>
    </log-quiet-control>
    <log-quiet-control>
        <log-type>raslog</log-type>
        <quiet-enabled>false</quiet-enabled>
        <days-of-week/>
    </log-quiet-control>
</Response>
```

Enabling Quiet Time for a Specific Log Type

The following example uses the PATCH request to enable quiet time for audit logs.

Structure

PATCH *<base_URI>*/running/brocade-logging/log-quiet-control/log-type/*log_type*/quiet-enabled/true

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-quiet-control/log-type/audit/quiet-enabled/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Disabling Quiet Time for a Specific Log Type

The following example uses the PATCH request to disable quiet time for audit logs.

Structure

PATCH *<base_URI>*/running/brocade-logging/log-quiet-control/log-type/*log_type*/quiet-enabled/false

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-quiet-control/log-type/audit/quiet-enabled/false
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring the Quiet Time

The following example uses the PATCH request to configure quiet time for a duration of forever for audit logs.

Structure

PATCH *<base_URI>*/running/brocade-logging/log-quiet-control

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-quiet-control
```

Request Body

```
<log-quiet-control>
    <log-type>audit</log-type>
    <days-of-week>forever</days-of-week>
</log-quiet-control>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring Quiet Time for a Specific Log Type with Start Time, End Time, and Days of Week Specified

The following example uses the PATCH request to enable quiet time for audit logs on Monday, Wednesday, and Friday at 7:00 AM and turns off at 1:00 PM.

Structure

PATCH *<base_URI>*/running/brocade-logging/log-quiet-control

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-quiet-control
```

Request Body

```
<log-quiet-control>
    <log-type>audit</log-type>
    <quiet-enabled>true</quiet-enabled>
    <start-time>0700</start-time>
    <end-time>1300</end-time>
    <days-of-week>
           <day>mon</day>
```

```
            <day>wed</day>
            <day>fri</day>
        </days-of-week>
    </log-quiet-control>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring the Syslog Facility

The following example uses the PATCH request to set the syslog facility to 'log_local2'.

Structure

PATCH *<base_URI>*/running/brocade-logging/log-setting

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-setting
```

Request Body

```
<log-setting>
    <syslog-facility-level>log_local2</syslog-facility-level>
</log-setting>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring the Keepalive Period

The following example uses the PATCH request to set the system keepalive period to '1'. The valid values are from 0 through 24; where 0 disables keepalive period.

Structure

PATCH *<base_URI>*/running/brocade-logging/log-setting/keep-alive-period/[0-24]

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-setting/keep-alive-period/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the Support FTP Settings

This example uses a GET request to view supportFTP settings.

Structure

GET <*base_URI*>/running/brocade-logging/supportftp

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/supportftp
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <supportftp>
        <host/>
        <user-name/>
        <password/>
        <remote-directory/>
        <auto-enabled>false</auto-enabled>
        <protocol>ftp</protocol>
        <connectivity-check-interval>0</connectivity-check-interval>
    </supportftp>
</Response>
```

Configuring Support FTP

This example uses a PATCH request to configure supportFTP.

Structure

PATCH <*base_URI*>/running/brocade-logging/supportftp

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/supportftp
```

Request Body

```
 <supportftp>
     <host>10.10.10.10</host>
     <user-name>Bob</user-name>
     <password>base64-pass</password>
     <remote-directory> </remote-directory>
     <auto-enabled>false</auto-enabled>
     <protocol>ftp</protocol>
     <connectivity-check-interval>0</connectivity-check-interval>
 </supportftp>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status in the headers.

Viewing Audit Logs

This example uses a GET request to view audit logs.

Structure

GET <*base_URI*>/running/brocade-logging/audit-log

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/audit-log
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <audit-log>
        <sequence-number>1</sequence-number>
        <time-stamp>2020-03-07T10:40:31-07:00</time-stamp>
        <message-id>RAS-3005</message-id>
        <message-text>CLI:  ## REST Operation: GET /brocade-maps/rule/name/defALL_32GSWL_SFPSFP_TEMP_85/
actions ##</message-text>
        <severity-level>info</severity-level>
        <event-class>cli</event-class>
        <user-name>admin</user-name>
        <role>admin</role>
        <ip-address>10.10.10.47</ip-address>
        <interface>ssh</interface>
        <application-name>CLI</application-name>
        <switch-user-friendly-name>BrocadeX6-4</switch-user-friendly-name>
        <fabric-id>0</fabric-id>
    </audit-log>
...
    <audit-log>
        <sequence-number>8192</sequence-number>
        <time-stamp>2020-03-09T14:55:11-06:00</time-stamp>
        <message-id>SEC-3022</message-id>
        <message-text>Event: logout, Status: success, Info: Successful logout by user [admin].</message-text>
        <severity-level>info</severity-level>
        <event-class>security</event-class>
        <user-name>admin</user-name>
        <role>admin</role>
        <ip-address>2020:100:2:f800:200:2dff:fe22:2e0c</ip-address>
        <interface>https</interface>
        <application-name>SANnavMP-2.1.0-f</application-name>
        <switch-user-friendly-name>66-243_X6-4_Venator</switch-user-friendly-name>
        <fabric-id>128</fabric-id>
    </audit-log>
</Response>
```

Viewing Error Logs

This example uses a GET request to view error logs.

Structure

GET <*base_URI*>/running/brocade-logging/error-log

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/error-log
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <error-log>
        <time-stamp>2020-03-02T10:01:37-07:00</time-stamp>
        <message-id>HAM-1004</message-id>
        <sequence-number>1</sequence-number>
        <severity-level>info</severity-level>
        <ffdc-generated-event>false</ffdc-generated-event>
        <fabric-id>0</fabric-id>
        <slot-id>1</slot-id>
        <switch-user-friendly-name>BrocadeX6-4</switch-user-friendly-name>
        <message-text>Processor rebooted - Reset.</message-text>
    </error-log>
...
    <error-log>
        <time-stamp>2020-03-09T14:42:56-06:00</time-stamp>
        <message-id>MAPS-1132</message-id>
        <sequence-number>723</sequence-number>
        <severity-level>info</severity-level>
        <ffdc-generated-event>false</ffdc-generated-event>
        <fabric-id>128</fabric-id>
        <slot-id>1</slot-id>
        <switch-user-friendly-name>66-243_X6-4_Venator</switch-user-friendly-name>
        <message-text>Monitoring on members 8/16 of type PORT is restarted.</message-text>
    </error-log>
</Response>
```

Viewing the Current Audit Log Configuration

The following example uses the GET request to view the audit log configuration.

Structure

```
GET <base_URI>/running/brocade-logging/audit
```

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/audit
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <audit>
        <audit-enabled>true</audit-enabled>
        <severity-level>info</severity-level>
        <filter-class-list>
            <filter-class>zone</filter-class>
            <filter-class>security</filter-class>
            <filter-class>configuration</filter-class>
            <filter-class>firmware</filter-class>
            <filter-class>fabric</filter-class>
            <filter-class>ls</filter-class>
            <filter-class>cli</filter-class>
            <filter-class>maps</filter-class>
        </filter-class-list>
    </audit>
</Response>
```

Enabling Audit Logging

The following example uses the PATCH request to enable audit logging.

Structure

```
PATCH <base_URI>/running/brocade-logging/audit/audit-enabled/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/audit/audit-enabled/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Disabling Audit Logging

The following example uses the PATCH request to disable audit logging.

Structure

```
PATCH <base_URI>/running/brocade-logging/audit/audit-enabled/false
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/audit/audit-enabled/false
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a "204 No Content” status appears in the header.

Configuring the Security Level

The following example uses the PATCH request to configure the security level to info. Only log messages of the specified severity level and higher display.

Structure

```
PATCH <base_URI>/running/brocade-logging/audit/severity-level/info
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/audit/severity-level/info
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Configuring the Filter Class

The following example uses the PATCH request to configure the filter class list to include zone, fabric, and MAPS.

Structure

```
PATCH <base_URI>/running/brocade-logging/audit/filter-class-list
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/audit/filter-class-list
```

Request Body

```
<filter-class-list>
     <filter-class>zone</filter-class>
     <filter-class>fabric</filter-class>
     <filter-class>maps</filter-class>
</filter-class-list>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Viewing the Current Syslog Configuration

The following example uses the GET request to display the current syslog configuration.

Structure

```
GET <base_URI>/running/brocade-logging/syslog-server
```

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/syslog-server
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <syslog-server>
        <server>10.10.10.11</server>
        <port>514</port>
        <secure-mode>false</secure-mode>
    </syslog-server>
    <syslog-server>
        <server>10.10.10.12</server>
        <port>514</port>
        <secure-mode>false</secure-mode>
    </syslog-server>
    <syslog-server>
        <server>10.10.10.13</server>
        <port>514</port>
        <secure-mode>false</secure-mode>
    </syslog-server>
</Response>
```

Configuring a Non-secure Syslog Server

The following example uses the POST request to configure a non-secure syslog server '17.14.15.16'.

Structure

```
POST <base_URI>/running/brocade-logging/syslog-server
```

URI

```
POST https://10.10.10.10/rest/running/brocade-logging/syslog-server
```

Request Body

```
<syslog-server>
    <server>17.14.15.16</server>
    <secure-mode>false</secure-mode>
</syslog-server>
```

Response Body

When the operation is successful, the response has a message body similar to the following, and a “201 Created” status appears in the headers.

Configuring a Secure Syslog Server

The following example uses the POST request to configure a secure syslog server '17.14.15.17'.

Structure

```
POST <base_URI>/running/brocade-logging/syslog-server
```

URI

```
POST https://10.10.10.10/rest/running/brocade-logging/syslog-server
```

Request Body

```
<syslog-server>
    <server>17.14.15.17</server>
    <port>5670</port>
    <secure-mode>true</secure-mode>
</syslog-server>
```

Response Body

When the operation is successful, the response has a message body similar to the following, and a “201 Created” status appears in the headers.

Removing a Syslog Server

The following example uses the DELETE request to remove syslog server '17.14.15.16'.

Structure

```
DELETE <base_URI>/running/brocade-logging/syslog-server/server/ip_address
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-logging/syslog-server/server/17.14.15.16
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content" status appears in the header.

Viewing all External RASlog Messages

The following example uses the GET request to displays all external RASlog messages, their status (enabled or disabled), their configured severity and their default severity.

Structure

```
GET <base_URI>/running/brocade-logging/raslog
```

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/raslog
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <raslog>
        <message-id>FCIP-1000</message-id>
        <current-severity>error</current-severity>
        <default-severity>error</default-severity>
        <message-enabled>true</message-enabled>
        <message-flooded>false</message-flooded>
    </raslog>
    <raslog>
        <message-id>FCIP-1001</message-id>
        <current-severity>critical</current-severity>
        <default-severity>critical</default-severity>
        <message-enabled>true</message-enabled>
        <message-flooded>false</message-flooded>
    </raslog>
    <raslog>
        <message-id>FCIP-1002</message-id>
        <current-severity>info</current-severity>
        <default-severity>info</default-severity>
        <message-enabled>true</message-enabled>
        <message-flooded>false</message-flooded>
    </raslog>
...
    <raslog>
        <message-id>CONF-1053</message-id>
        <current-severity>info</current-severity>
        <default-severity>info</default-severity>
        <message-enabled>true</message-enabled>
        <message-flooded>false</message-flooded>
    </raslog>
    <raslog>
        <message-id>CONF-1054</message-id>
        <current-severity>info</current-severity>
        <default-severity>info</default-severity>
        <message-enabled>true</message-enabled>
        <message-flooded>false</message-flooded>
    </raslog>
    <raslog>
        <message-id>CONF-1055</message-id>
        <current-severity>info</current-severity>
        <default-severity>info</default-severity>
        <message-enabled>true</message-enabled>
        <message-flooded>false</message-flooded>
    </raslog>
</Response>
```

Viewing the Status of a Specific RASlog Message

The following example uses the GET request to view the status of a specific RASlog message 'AUTH-1002'.

Structure

```
GET <base_URI>/running/brocade-logging/raslog/message-id/message_id
```

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/raslog/message-id/<message_id>
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<xml version="1.0">
<Response>
    <raslog>
        <message-id>AUTH-1002</message-id>
        <message-enabled>true</message-enabled>
        <message-flooded>false</message-flooded>
        <current-severity>error</current-severity>
        <default-severity>error</default-severity>
    </raslog>
</Response>
```

Enabling Logging for a Specific RASlog Message

The following example uses the PATCH request to enable logging for a specific RASlog message 'AUTH-1002'.

Structure

```
PATCH <base_URI>/running/brocade-logging/raslog/message-id/message_id/message-enabled/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/raslog/message-id/AUTH-1002/message-enabled/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Disabling Logging for a Specific RASlog Message

The following example uses the PATCH request to disable logging for a specific RASlog message 'AUTH-1002'.

Structure

```
PATCH <base_URI>/running/brocade-logging/raslog/message-id/message_id/message-enabled/false
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/raslog/message-id/AUTH-1002/message-enabled/false
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Enabling Syslog for a Specific Internal RASlog Message

The following example uses the PATCH request to enable an internal RASlog message 'CONF-5000' to be sent to syslog (this is done per instruction from support). The syslog-enabled parameter is only supported for internal RASlog messages. Internal RASlog messages have an ID number equal to or greater than 5000.

Structure

```
PATCH <base_URI>/running/brocade-logging/raslog/message-id/message_id/syslog-enabled/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/raslog/message-id/CONF-5000/syslog-enabled/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Disabling Syslog for a Specific RASlog Message

The following example uses the PATCH request to disable an internal RASlog message 'CONF-5000' for syslog (this is done per instruction from support). The syslog-enabled parameter is only supported for internal RASlog messages. Internal RASlog messages have an ID number equal to or greater than 5000.

Structure

```
PATCH <base_URI>/running/brocade-logging/raslog/message-id/message_id/syslog-enabled/false
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/raslog/message-id/CONF-5000/syslog-enabled/false
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring Settings for a Specific External RASlog Message

The following example uses the PATCH request to configure the following settings for a specific external RASlog message 'AUTH-1002':

- message-enabled = true
- default-severity = info

External RASlog messages have an ID number equal to or less than 4999.

Structure

```
PATCH <base_URI>/running/brocade-logging/raslog
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/raslog
```

Request Body

```
<raslog>
    <message-id>AUTH-1002</message-id>
    <message-enabled>true</message-enabled>
    <current-severity>info</current-severity>
</raslog>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring Settings for a Specific Internal RASlog Message

The following example uses the PATCH request to configure the following settings for a specific Internal RASlog message 'CONF-5000':

- default-severity = error
- syslog-enabled = true

The syslog-enabled parameter is only supported for internal RASlog messages. Internal RASlog messages have an ID number equal to or greater than 5000.

Structure

```
PATCH <base_URI>/running/brocade-logging/raslog
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/raslog
```

Request Body

```
<raslog>
    <message-id>CONF-5000</message-id>
    <syslog-enabled>true</syslog-enabled>
    <current-severity>error</current-severity>
</raslog>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing a List of the Current RASlog Modules

The following example uses the GET request to view a list of all RASlog modules.

Structure

```
GET <base_URI>/running/brocade-logging/raslog-module
```

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/raslog-module
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <raslog-module>
        <module-id>UT</module-id>
    </raslog-module>
    <raslog-module>
        <module-id>TRCE</module-id>
    </raslog-module>
    <raslog-module>
        <module-id>KTRC</module-id>
    </raslog-module>
    ...
    <raslog-module>
        <module-id>UCID</module-id>
    </raslog-module>
    <raslog-module>
        <module-id>AMPM</module-id>
    </raslog-module>
    <raslog-module>
        <module-id>BCMG</module-id>
    </raslog-module>
</Response>
```

Enabling Logging for all Messages in a RASlog Module Group

The following example uses the PATCH request to enable logging for all messages in the RASlog module group 'UT'.

Structure

```
PATCH <base_URI>/running/brocade-logging/raslog-module/module_id/log-enabled/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/raslog-module/UT/log-enabled/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Disabling Logging for all Messages in a RASlog Module Group

The following example uses the PATCH request to disable logging for all RASlog messages in the RASlog module group 'TRCE'.

Structure

```
PATCH <base_URI>/running/brocade-logging/raslog-module/module_id/log-enabled/false
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/raslog-module/TRCE/log-enabled/false
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the Quiet Time Configuration

The following example uses the GET request to display the current quiet time configuration.

Structure

```
GET <base_URI>/running/brocade-logging/log-quiet-control
```

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/log-quiet-control
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<Response>
    <log-quiet-control>
        <log-type>audit</log-type>
        <quiet-enabled>false</quiet-enabled>
        <days-of-week/>
    </log-quiet-control>
    <log-quiet-control>
        <log-type>raslog</log-type>
        <quiet-enabled>false</quiet-enabled>
        <days-of-week/>
    </log-quiet-control>
</Response>
```

Enabling Quiet Time for a Specific Log Type

The following example uses the PATCH request to enable quiet time for audit logs.

Structure

```
PATCH <base_URI>/running/brocade-logging/log-quiet-control/log-type/log_type/quiet-enabled/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-quiet-control/log-type/audit/quiet-enabled/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Disabling Quiet Time for a Specific Log Type

The following example uses the PATCH request to disable quiet time for audit logs.

Structure

```
PATCH <base_URI>/running/brocade-logging/log-quiet-control/log-type/log_type/quiet-enabled/false
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-quiet-control/log-type/audit/quiet-enabled/false
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring the Quiet Time

The following example uses the PATCH request to configure quiet time for a duration of forever for audit logs.

Structure

```
PATCH <base_URI>/running/brocade-logging/log-quiet-control
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-quiet-control
```

Request Body

```
<log-quiet-control>
    <log-type>audit</log-type>
    <days-of-week>forever</days-of-week>
</log-quiet-control>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring Quiet Time for a Specific Log Type with Start Time, End Time, and Days of Week Specified

The following example uses the PATCH request to enable quiet time for audit logs on Monday, Wednesday, and Friday at 7:00 AM and turns off at 1:00 PM.

Structure

```
PATCH <base_URI>/running/brocade-logging/log-quiet-control
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-quiet-control
```

Request Body

```
<log-quiet-control>
    <log-type>audit</log-type>
    <quiet-enabled>true</quiet-enabled>
    <start-time>0700</start-time>
    <end-time>1300</end-time>
    <days-of-week>
           <day>mon</day>
           <day>wed</day>
           <day>fri</day>
    </days-of-week>
</log-quiet-control>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring the Syslog Facility

The following example uses the PATCH request to set the syslog facility to 'log_local2'.

Structure

```
PATCH <base_URI>/running/brocade-logging/log-setting
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-setting
```

Request Body

```
<log-setting>
    <syslog-facility-level>log_local2</syslog-facility-level>
</log-setting>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring the Keepalive Period

The following example uses the PATCH request to set the system keepalive period to '1'. The valid values are from 0 through 24; where 0 disables keepalive period.

Structure

```
PATCH <base_URI>/running/brocade-logging/log-setting/keep-alive-period/[0-24]
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-setting/keep-alive-period/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the Support FTP Settings

This example uses a GET request to view supportFTP settings.

Structure

```
GET <base_URI>/running/brocade-logging/supportftp
```

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/supportftp
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <supportftp>
        <host/>
        <user-name/>
        <password/>
        <remote-directory/>
        <auto-enabled>false</auto-enabled>
        <protocol>ftp</protocol>
        <connectivity-check-interval>0</connectivity-check-interval>
    </supportftp>
</Response>
```

Configuring Support FTP

This example uses a PATCH request to configure supportFTP.

Structure

```
PATCH <base_URI>/running/brocade-logging/supportftp
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-logging/supportftp
```

Request Body

```
<supportftp>
     <host>10.10.10.10</host>
     <user-name>Bob</user-name>
     <password>base64-pass</password>
     <remote-directory> </remote-directory>
     <auto-enabled>false</auto-enabled>
     <protocol>ftp</protocol>
     <connectivity-check-interval>0</connectivity-check-interval>
 </supportftp>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status in the headers.

Viewing Audit Logs

This example uses a GET request to view audit logs.

Structure

```
GET <base_URI>/running/brocade-logging/audit-log
```

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/audit-log
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <audit-log>
        <sequence-number>1</sequence-number>
        <time-stamp>2020-03-07T10:40:31-07:00</time-stamp>
        <message-id>RAS-3005</message-id>
        <message-text>CLI:  ## REST Operation: GET /brocade-maps/rule/name/defALL_32GSWL_SFPSFP_TEMP_85/
actions ##</message-text>
        <severity-level>info</severity-level>
        <event-class>cli</event-class>
        <user-name>admin</user-name>
        <role>admin</role>
        <ip-address>10.10.10.47</ip-address>
        <interface>ssh</interface>
        <application-name>CLI</application-name>
        <switch-user-friendly-name>BrocadeX6-4</switch-user-friendly-name>
        <fabric-id>0</fabric-id>
    </audit-log>
```

```
...
    <audit-log>
        <sequence-number>8192</sequence-number>
        <time-stamp>2020-03-09T14:55:11-06:00</time-stamp>
        <message-id>SEC-3022</message-id>
        <message-text>Event: logout, Status: success, Info: Successful logout by user [admin].</message-text>
        <severity-level>info</severity-level>
        <event-class>security</event-class>
        <user-name>admin</user-name>
        <role>admin</role>
        <ip-address>2020:100:2:f800:200:2dff:fe22:2e0c</ip-address>
        <interface>https</interface>
        <application-name>SANnavMP-2.1.0-f</application-name>
        <switch-user-friendly-name>66-243_X6-4_Venator</switch-user-friendly-name>
        <fabric-id>128</fabric-id>
    </audit-log>
</Response>
```

Viewing Error Logs

This example uses a GET request to view error logs.

Structure

```
GET <base_URI>/running/brocade-logging/error-log
```

URI

```
GET https://10.10.10.10/rest/running/brocade-logging/error-log
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <error-log>
        <time-stamp>2020-03-02T10:01:37-07:00</time-stamp>
        <message-id>HAM-1004</message-id>
        <sequence-number>1</sequence-number>
        <severity-level>info</severity-level>
        <ffdc-generated-event>false</ffdc-generated-event>
        <fabric-id>0</fabric-id>
        <slot-id>1</slot-id>
        <switch-user-friendly-name>BrocadeX6-4</switch-user-friendly-name>
        <message-text>Processor rebooted - Reset.</message-text>
    </error-log>
...
    <error-log>
        <time-stamp>2020-03-09T14:42:56-06:00</time-stamp>
        <message-id>MAPS-1132</message-id>
```

```
        <sequence-number>723</sequence-number>
        <severity-level>info</severity-level>
        <ffdc-generated-event>false</ffdc-generated-event>
        <fabric-id>128</fabric-id>
        <slot-id>1</slot-id>
        <switch-user-friendly-name>66-243_X6-4_Venator</switch-user-friendly-name>
        <message-text>Monitoring on members 8/16 of type PORT is restarted.</message-text>
    </error-log>
</Response>
```

brocade-management-ip-interface

This module enables you to manage the IP configurations.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-management-ip-interface.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-management-ip-interface
  +--rw brocade-management-ip-interface
     +--rw management-ip-interface
     |  +--rw static-ip-addresses
     |  |  +--rw ip-address*   union
     |  +--ro stateful-ip-addresses
     |  |  +--ro ip-address*   union
     |  +--ro stateless-ipv6-addresses
     |  |  +--ro ip-address*   union
     |  +--rw static-ip-gateways
     |  |  +--rw ip-gateway*   fru:ip-gateway-type
     |  +--ro stateless-ipv6-gateways
     |  |  +--ro ip-gateway*   union
     |  +--ro dhcp-gateway?               union
     |  +--rw static-subnet-mask?         union
     |  +--ro dhcp-subnet-mask?           union
     |  +--rw dhcp-enabled?               boolean
     |  +--rw dhcpv6-enabled?             boolean
     |  +--rw ipv6-ddns-enabled?          boolean
     |  +--rw ipv6-auto-enabled?          boolean
     +--ro management-interface-lldp-neighbor* [cp-name physical-interface]
     |  +--ro cp-name                   fibrechannel:cp-name-type
     |  +--ro physical-interface        fibrechannel:physical-interface-type
     |  +--ro dead-interval?            uint32
     |  +--ro remaining-life?           uint32
     |  +--ro remote-interface-name?    string
     |  +--ro chassis-id?               string
     |  +--ro system-name?              string
     +--ro management-interface-lldp-statistics* [cp-name physical-interface]
        +--ro cp-name                fibrechannel:cp-name-type
        +--ro physical-interface     fibrechannel:physical-interface-type
        +--ro out-frames?            ietf:zero-based-counter64
        +--ro in-frames?             ietf:zero-based-counter64
        +--ro frames-aged-out?       ietf:zero-based-counter64
        +--ro frames-discarded?      ietf:zero-based-counter64
        +--ro error-frames?          ietf:zero-based-counter64
        +--ro tlv-discarded?         ietf:zero-based-counter64
        +--ro tlv-unrecognized?      ietf:zero-based-counter64
```

Examples

This section provides basic examples for using the brocade-management-ip-interface module.

Viewing the LLDP Statistics for the Management Interface

This example uses a GET request to display the LLDP statistics for the management interface.

Structure

```
GET <base_URI>/running/brocade-management-ip-interface/management-interface-lldp-statistics
```

URI

```
GET https://10.10.10.10/rest/running/brocade-management-ip-interface/management-interface-lldp-statistics
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <management-interface-lldp-statistics>
        <cp-name>cp0</cp-name>
        <physical-interface>eth0</physical-interface>
        <out-frames>338</out-frames>
        <frames-aged-out>0</frames-aged-out>
        <frames-discarded>0</frames-discarded>
        <error-frames>0</error-frames>
        <in-frames>339</in-frames>
        <tlv-discarded>0</tlv-discarded>
        <tlv-unrecognized>0</tlv-unrecognized>
    </management-interface-lldp-statistics>
    <management-interface-lldp-statistics>
        <cp-name>cp1</cp-name>
        <physical-interface>eth0</physical-interface>
        <out-frames>156</out-frames>
        <frames-aged-out>0</frames-aged-out>
        <frames-discarded>0</frames-discarded>
        <error-frames>0</error-frames>
        <in-frames>157</in-frames>
        <tlv-discarded>0</tlv-discarded>
        <tlv-unrecognized>0</tlv-unrecognized>
    </management-interface-lldp-statistics>
</Response>
```

Viewing the LLDP Neighbor Devices for the Management Interface

This example uses a GET request to display the LLDP neighbor devices for the management interface.

Structure

```
GET <base_URI>/running/brocade-management-ip-interface/management-interface-lldp-neighbor
```

URI

```
GET https://10.10.10.10/rest/running/brocade-management-ip-interface/management-interface-lldp-neighbor
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <management-interface-lldp-neighbor>
        <cp-name>cp0</cp-name>
        <physical-interface>eth0</physical-interface>
        <dead-interval>120</dead-interval>
        <remaining-life>96</remaining-life>
        <remote-interface-name>Gi1/0/31</remote-interface-name>
        <chassis-id>00fc.ba07.c880</chassis-id>
        <system-name>BRM-3D2-C06-RS1.nw-am.broadcom.net</system-name>
    </management-interface-lldp-neighbor>
    <management-interface-lldp-neighbor>
        <cp-name>cp1</cp-name>
        <physical-interface>eth0</physical-interface>
        <dead-interval>120</dead-interval>
        <remaining-life>98</remaining-life>
        <remote-interface-name>Gi1/0/2</remote-interface-name>
        <chassis-id>00fc.ba07.c880</chassis-id>
        <system-name>BRM-3D2-C06-RS1.nw-am.broadcom.net</system-name>
    </management-interface-lldp-neighbor>
</Response>
```

brocade-maps

This module provides a detailed view of the Monitoring and Alerting Policy Suite (MAPS) features available in the Fabric OS REST API. It assumes a knowledge of MAPS as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS MAPS User Guide*.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-maps.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-maps
  +--rw brocade-maps
     +--ro switch-status-policy-report
     |  +--ro switch-health?                    maps-types:ssp-state-type
     |  +--ro power-supply-health?              maps-types:ssp-state-type
     |  +--ro fan-health?                       maps-types:ssp-state-type
     |  +--ro wwn-health?                       maps-types:ssp-state-type
  {fibrechannel:fibrechannel_chassis_platform}?
     |  +--ro temperature-sensor-health?        maps-types:ssp-state-type
     |  +--ro ha-health?                        maps-types:ssp-state-type
  {fibrechannel:fibrechannel_chassis_platform}?
     |  +--ro control-processor-health?         maps-types:ssp-state-type
  {fibrechannel:fibrechannel_chassis_platform}?
     |  +--ro core-blade-health?                maps-types:ssp-state-type
  {fibrechannel:fibrechannel_chassis_platform}?
     |  +--ro blade-health?                     maps-types:ssp-state-type
  {fibrechannel:fibrechannel_chassis_platform}?
     |  +--ro flash-health?                     maps-types:ssp-state-type
     |  +--ro marginal-port-health?             maps-types:ssp-state-type
     |  +--ro faulty-port-health?               maps-types:ssp-state-type
     |  +--ro missing-sfp-health?               maps-types:ssp-state-type
     |  +--ro error-port-health?                maps-types:ssp-state-type
     |  +--ro expired-certificate-health?       maps-types:ssp-state-type
     |  +--ro airflow-mismatch-health?          maps-types:ssp-state-type
     |  +--ro marginal-sfp-health?              maps-types:ssp-state-type
     |  o--ro trusted-fos-certificate-health?   maps-types:ssp-state-type
     +--ro system-resources
     |  +--ro cpu-usage?            uint32
     |  +--ro memory-usage?         uint32
     |  +--ro total-memory?         uint32
     |  +--ro flash-usage?          uint32
     |  +--ro free-kernel-memory?   uint32
     +--rw paused-cfg* [group-type]
     |  +--rw group-type    enumeration
     |  +--rw members
     |     +--rw member*   string
     +--rw group* [name]
     |  +--rw name               maps-types:maps-group-name-type
     |  +--rw group-type         maps-types:maps-group-type-type
     |  +--rw group-feature?     maps-types:maps-group-feature-type
     |  +--rw feature-pattern?   string
```

```
|  +--ro is-predefined?     boolean
|  +--ro is-modifiable?     boolean
|  +--rw members
|     +--rw member*   string
+--rw maps-config
|  +--rw actions
|  |  +--rw action*   maps-types:maps-generic-action-type
|  +--rw decommission-cfg?         enumeration
|  +--rw recipient-address-list
|  |  +--rw recipient-address*   string
|  +--rw sender-address?           string
|  +--rw domain-name?              string
|  +--rw relay-ip-address?         union
|  x--rw quiet-time?               uint32
|  +--rw test-email
|  |  +--rw subject?   string
|  |  +--rw body?      string
|  +--rw f-port-fpi-profile?       maps-types:maps-policy-name-type
|  +--rw e-port-fpi-profile?       maps-types:maps-policy-name-type
|  +--rw notification?             maps-types:notification-type
+--ro dashboard-rule* []
|  +--ro category?           maps-types:maps-dashboard-category-type
|  +--ro name?               maps-types:maps-rule-name-type
|  +--ro triggered-count?    uint32
|  +--ro time-stamp?         yang:date-and-time
|  +--ro repetition-count?   uint32
|  +--ro objects
|     +--ro object*   string
+--rw dashboard-misc
|  +--ro maps-start-time?            yang:date-and-time
|  +--rw clear-data?                 boolean
|  +--ro congestion-db-start-time?   yang:date-and-time
|  +--ro fenced-ports
|  |  +--ro port*   string
|  +--ro decommissioned-ports
|  |  +--ro port*   string
|  +--ro quarantined-ports
|  |  +--ro port*   string
|  +--ro fenced-circuits
|  |  +--ro circuit*   string
|  +--ro pid-it-flows
|     +--ro pid*   string
+--ro credit-stall-dashboard* []
|  +--ro slot-port?                fibrechannel:slot-port-name-type
|  +--ro timestamp?                yang:date-and-time
|  +--ro state?                    maps-types:credit-stall-state-type
|  +--ro frame-loss-state-count?   uint8
|  +--ro perf-impact-state-count?  uint8
|  +--ro medium-state-count?       uint8
|  +--ro low-state-count?          uint8
|  +--ro info-state-count?         uint8
|  +--ro frequency?                uint8
+--ro oversubscription-dashboard* []
```

```
|  +--ro slot-port?   fibrechannel:slot-port-name-type
|  +--ro timestamp?   yang:date-and-time
|  +--ro state?       maps-types:oversubscription-state-type
|  +--ro frequency?   uint8
+--rw rule* [name]
|  +--rw name                    maps-types:maps-rule-name-type
|  +--rw is-rule-on-rule         maps-types:maps-rule-type
|  +--rw monitoring-system       maps-types:maps-monitoring-system-type
|  +--rw time-base               maps-types:maps-time-base-type
|  +--rw logical-operator        maps-types:maps-logical-operator-type
|  +--rw threshold-value         maps-types:threshold-value-type
|  +--rw group-name?             maps-types:maps-group-name-type
|  +--rw actions
|  |  +--rw action*   maps-types:maps-generic-action-type
|  +--ro is-predefined?          boolean
|  +--rw event-severity?         maps-types:maps-event-severity-type
|  +--rw toggle-time?            uint32
|  x--rw quiet-time?             uint32
|  x--rw quiet-time-clear?       boolean
|  +--rw un-quarantine-timeout?  uint32
|  o--rw un-quarantine-clear?    boolean
+--rw maps-policy* [name]
|  +--rw name                   maps-types:maps-policy-name-type
|  +--rw rule-list
|  |  +--rw rule*   maps-types:maps-rule-name-type
|  +--ro system-rules
|  |  +--ro rule*   maps-types:maps-rule-name-type
|  +--ro unmonitored-rules
|  |  +--ro rule*   maps-types:maps-rule-name-type
|  +--rw is-active-policy?      boolean
|  +--ro is-predefined-policy?  boolean
|  +--ro deleted-rules
|     +--ro rule*   maps-types:maps-rule-name-type
+--ro monitoring-system-matrix* [monitoring-system group-type]
|  +--ro monitoring-system                maps-types:maps-monitoring-system-type
|  +--ro group-type                       maps-types:maps-group-type-type
|  +--ro dashboard-category?              maps-types:maps-dashboard-category-type
|  +--ro base-time-bases
|  |  +--ro time-base*   maps-types:maps-time-base-type
|  +--ro rule-on-rule-time-bases
|  |  +--ro rule-on-rule-time-base*   maps-types:maps-time-base-type
|  +--ro is-read-only?                    boolean
|  +--ro monitored-logical-switch?        enumeration
|  +--ro is-rule-on-rule-supported?       boolean
|  x--ro is-quiet-time-supported?         boolean
|  x--ro minimum-quiet-time?              uint32
|  +--ro monitoring-type?                 maps-types:monitoring-type-type
|  +--ro data-type?                       maps-types:data-type-type
|  +--ro description?                     string
|  +--ro actions
|  |  +--ro action*   maps-types:maps-generic-action-type
|  +--ro unit?                            maps-types:maps-data-unit
|  +--ro data-range?                      string
```

```
|  +--ro logical-operators
|  |  +--ro logical-operator*   maps-types:maps-logical-operator-type
|  +--ro adaptive-notification-supported?   boolean
+--rw fpi-profile* [name]
|  +--rw name                     maps-types:maps-policy-name-type
|  +--ro predefined-profile?      boolean
|  +--rw performance-impact
|  |  +--rw transmit-queue-latency          uint32
|  |  +--rw credit-zero-percentage-1-sec    uint32
|  |  +--rw credit-zero-percentage-5-sec    uint32
|  |  +--rw credit-zero-percentage-10-sec   uint32
|  +--rw frame-loss
|  |  +--rw transmit-queue-latency    uint32
|  +--rw port-oversubscription
|     +--rw transmit-percentage?   uint32
+--ro dashboard-history* []
|  +--ro category?              maps-types:port-type
|  +--ro date?                  maps-types:maps-db-date-type
|  +--ro crc-errors
|  |  +--ro port-data*   string
|  +--ro itw-errors
|  |  +--ro port-data*   string
|  x--ro port-loss
|  |  +--ro port-data*   string
|  +--ro loss-of-sync
|  |  +--ro port-data*   string
|  +--ro link-failure
|  |  +--ro port-data*   string
|  +--ro signal-loss
|  |  +--ro port-data*   string
|  +--ro protocol-errors
|  |  +--ro port-data*   string
|  +--ro state-change
|  |  +--ro port-data*   string
|  +--ro link-reset
|  |  +--ro port-data*   string
|  +--ro c3-tx-to
|  |  +--ro port-data*   string
|  +--ro rx-perf
|  |  +--ro port-data*   string
|  +--ro tx-perf
|  |  +--ro port-data*   string
|  +--ro utilization
|  |  +--ro port-data*   string
|  +--ro bottleneck-duration
|     +--ro port-data*   string
+--ro maps-device-login* [node-wwn]
|  +--ro node-wwn       fibrechannel:wwn-type
|  +--ro state?         maps-types:port-group-state-type
|  +--ro master-port?   string
|  +--ro ports
|     +--ro port*   string
+--ro backend-ports-history* []
```

```
|  +--ro category?          maps-types:port-type
|  +--ro date?              maps-types:maps-db-date-type
|  +--ro crc-errors
|  |  +--ro port-data*   string
|  +--ro itw-errors
|  |  +--ro port-data*   string
|  +--ro link-reset
|  |  +--ro port-data*   string
|  +--ro bad-os
|  |  +--ro port-data*   string
|  +--ro long-frame
|  |  +--ro port-data*   string
|  +--ro truncated-frame
|     +--ro port-data*   string
+--ro gigabit-ethernet-ports-history* []
|  +--ro category?         maps-types:port-type
|  +--ro date?             maps-types:maps-db-date-type
|  +--ro ge-crc-errors
|  |  +--ro port-data*   string
|  +--ro ge-signal-loss
|     +--ro port-data*   string
+--ro quarantined-devices* [port-index port-id]
   +--ro port-index      fibrechannel:port-number-type
   +--ro port-id         fibrechannel:fcid-hex-string-type
   +--ro port-wwn?       fibrechannel:wwn-type
   +--ro locality?       enumeration
   +--ro port-name?      string
   +--ro zone-aliases
      +--ro alias-name*   zone:zoning-name-type
```

Examples

This section provides examples for the brocade-maps module.

It assumes a knowledge of Monitoring and Alerting Policy Suite as performed in Fabric OS. For information on that topic, refer to the *Brocade Monitoring and Alerting Policy Suite Configuration Guide*.

Viewing the Switch Status Policy Report

The following example uses the GET request to display the Switch Status Policy report.

Structure

```
GET <base_URI>/running/brocade-maps/switch-status-policy-report
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/switch-status-policy-report
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <switch-status-policy-report>
        <switch-health>healthy</switch-health>
        <power-supply-health>healthy</power-supply-health>
        <fan-health>healthy</fan-health>
        <wwn-health>healthy</wwn-health>
        <temperature-sensor-health>healthy</temperature-sensor-health>
        <ha-health>healthy</ha-health>
        <control-processor-health>healthy</control-processor-health>
        <core-blade-health>healthy</core-blade-health>
        <blade-health>healthy</blade-health>
        <flash-health>healthy</flash-health>
        <marginal-port-health>healthy</marginal-port-health>
        <faulty-port-health>healthy</faulty-port-health>
        <missing-sfp-health>healthy</missing-sfp-health>
        <error-port-health>healthy</error-port-health>
        <expired-certificate-health>healthy</expired-certificate-health>
        <airflow-mismatch-health>healthy</airflow-mismatch-health>
        <marginal-sfp-health>healthy</marginal-sfp-health>
        <trusted-fos-cert-health>healthy</trusted-fos-cert-health>
    </switch-status-policy-report>
</Response>
```

Viewing System Resources

The following example uses the GET request to display the system resources.

Structure

```
GET <base_URI>/running/brocade-maps/system-resources
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/system-resources
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <system-resources>
        <cpu-usage>2</cpu-usage>
        <memory-usage>8</memory-usage>
        <total-memory>15003372</total-memory>
        <flash-usage>13</flash-usage>
    </system-resources>
</Response>
```

Viewing the Paused Configuration

The following example uses the GET request to display the current paused configurations. To display a specific paused configuration, specify the group name in the request.

For example, the following shows specifying the group name (fc-port) in the request:

```
GET https://10.10.10.10/rest/running/brocade-maps/paused-cfg/group-type/fc-port
```

Structure

```
GET <base_URI>/running/brocade-maps/paused-cfg
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/paused-cfg
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <paused-cfg>
        <group-type>fc-port</group-type>
        <members>
            <member>8/9</member>
            <member>8/15</member>
            <member>8/16</member>
```

```
            <member>8/17</member>
            <member>8/18</member>
            <member>6/23</member>
        </members>
    </paused-cfg>
</Response>
```

Pausing MAPS Monitoring

The following example uses the POST request to pause MAPS monitoring of ports 8/9, 8/17, 8/18, and 6/23 in the fc-port group.

Structure

```
POST <base_URI>/running/brocade-maps/paused-cfg/
```

URI

```
POST https://10.10.10.10/rest/running/brocade-maps/paused-cfg/
```

Request Body

```
<paused-cfg>
    <group-type>fc-port</group-type>
    <members>
        <member>8/9</member>
        <member>8/17</member>
        <member>8/18</member>
        <member>6/23</member>
    </members>
</paused-cfg>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Resuming MAPS Monitoring

The following example uses the DELETE request to resume MAPS monitoring of ports 8/17, 8/18, and 6/23 in the fc-port group by deleting them from the paused configuration.

Structure

```
DELETE <base_URI>/running/brocade-maps/paused-cfg/
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-maps/paused-cfg/
```

Request Body

```
<paused-cfg>
    <group-type>fc-port</group-type>
    <members>
        <member>8/17</member>
        <member>8/18</member>
```

```
        <member>6/23</member>
    </members>
</paused-cfg>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing MAPS Groups

The following example uses the GET request to display the configured MAPS groups.

Structure

```
GET <base_URI>/running/brocade-maps/group
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/group
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <group>
        <name>ALL_PORTS</name>
        <group-type>fc-port</group-type>
        <members>
            <member>5/0</member>
            <member>5/1</member>
            <member>5/2</member>
            <member>5/3</member>
            <member>5/4</member>
            <member>5/5</member>
            <member>5/6</member>
...
     <group>
        <name>ALL_FCOE_100G_SR4_QSFP</name>
        <group-type>sfp</group-type>
        <members>
            <member/>
        </members>
        <is-predefined>true</is-predefined>
        <is-augmentable>false</is-augmentable>
    </group>
    <group>
        <name>ALL_OTHER_F_PORTS_1</name>
        <group-type>fc-port</group-type>
        <members>
            <member>8/5</member>
```

```
        </members>
        <is-predefined>false</is-predefined>
        <is-augmentable>false</is-augmentable>
    </group>
</Response>
```

Configuring a New MAPS Group

The following example uses the POST request to configure a new group ALL_OTHER_F_PORTS_1 with one port (port 5/8).

Structure

```
POST <base_URI>/running/brocade-maps/group
```

URI

```
POST https://10.10.10.10/rest/running/brocade-maps/group
```

Request Body

```
<group>
    <name>ALL_OTHER_F_PORTS_1</name>
    <group-type>fc-port</group-type>
    <members>
        <member>5/8</member>
    </members>
</group>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Modifying an Existing MAPS Group

The following example uses the PATCH request to modify the existing group ALL_FCOE_100G_SR4_QSFP with two ports 5/8 and 5/17.

Structure

```
 PATCH <base_URI>/running/brocade-maps/group
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-maps/group
```

Request Body

```
<group>
    <name>ALL_FCOE_100G_SR4_QSFP</name>
    <group-type>sfp</group-type>
    <members>
        <member>5/8</member>
        <member>5/17</member>
    </members>
</group>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Changing the MAPS Configuration

The following example uses the PATCH request to edit the current MAPS configuration by adding an e-mail recipient 'test2@broadcom.com' and configuring the quiet time to 2 hours and the F_Port FPI profile name to new-fport-fpi.

Structure

```
PATCH <base_URI>/running/brocade-maps/maps-config
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-maps/maps-config
```

Request Body

```
<maps-config>
    <recipient-address-list>
        <recipient-address>test1@broadcom.com</recipient-address>
        <recipient-address>test2@broadcom.com</recipient-address>
    </recipient-address-list>
    <quiet-time>2</quiet-time>
    <actions>
        <action>raslog</action>
        <action>e-mail</action>
        <action>sw-critical</action>
        <action>sw-marginal</action>
    </actions>
    <sender-address>sqa@brocade.com</sender-address>
    <decommission-cfg>impair</decommission-cfg>
    <f-port-fpi-profile>new-fport-fpi</f-port-fpi-profile>
</maps-config>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the MAPS Dashboard

The following example uses the GET request to display the MAPS dashboard for virtual fabric ID 77.

Note that a dashboard rule captures only the top five threshold violations (objects).

Structure

```
GET <base_URI>/running/brocade-maps/dashboard-rule?vf-id=<VF_ID>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/dashboard-rule?vf-id=77
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <dashboard-rule>
        <category>port-health</category>
        <triggered-count>10</triggered-count>
        <name>defALL_16GSWL_SFPCURRENT_2</name>
        <time-stamp>2020-05-18T22:36:00-06:00</time-stamp>
        <repetition-count>2</repetition-count>
        <objects>
            <object>SFP 1:0</object>
            <object>SFP 1:0</object>
        </objects>
    </dashboard-rule>
    <dashboard-rule>
        <category>port-health</category>
        <triggered-count>10</triggered-count>
        <name>defALL_32GSWL_SFPTXP_251</name>
        <time-stamp>2020-05-18T22:36:00-06:00</time-stamp>
        <repetition-count>4</repetition-count>
        <objects>
            <object>SFP 22:0</object>
            <object>SFP 17:0</object>
            <object>SFP 22:0</object>
            <object>SFP 17:0</object>
            <object>SFP 22:0</object>
        </objects>
    </dashboard-rule>
    <dashboard-rule>
        <category>port-health</category>
        <triggered-count>10</triggered-count>
        <name>defALL_32GSWL_SFPCURRENT_1</name>
        <time-stamp>2020-05-18T22:36:00-06:00</time-stamp>
        <repetition-count>4</repetition-count>
        <objects>
            <object>SFP 22:0</object>
            <object>SFP 17:0</object>
            <object>SFP 22:0</object>
            <object>SFP 17:0</object>
            <object>SFP 22:0</object>
        </objects>
    </dashboard-rule>
    <dashboard-rule>
        <category>fabric-state-changes</category>
        <triggered-count>5</triggered-count>
        <name>defSWITCHFLOGI_4</name>
        <time-stamp>2020-05-18T22:31:30-06:00</time-stamp>
        <repetition-count>5</repetition-count>
        <objects>
            <object>Switch:11</object>
            <object>Switch:5</object>
```

```
            <object>Switch:10</object>
            <object>Switch:5</object>
            <object>Switch:5</object>
        </objects>
    </dashboard-rule>
</Response>
```

The response body contains the same content that you would see in the Rules Affecting Health section of the MAPS dashboard generated from the CLI (see below).

```
switch:admin> mapsdb --show

1 Dashboard Information:
========================

DB start time:                  Mon May 18 22:17:15 2020
Active policy:                  dflt_aggressive_policy
Configured Notifications:       SW_CRITICAL,SW_MARGINAL,SFP_MARGINAL,FPIN
Fenced Ports :                  None
Decommissioned Ports :          None
Fenced circuits :               N/A
Quarantined Ports :             None
Top Zoned PIDs <pid(it-flows)>: 0xe81600(5) 0xe80a00(5)

2 Switch Health Report:
=======================

Current Switch Policy Status: HEALTHY


3.1 Summary Report:
===================

Category                 |Today                     |Last 7 days               |
-------------------------------------------------------------------------------
Port Health              |No Errors                 |Out of operating range    |
BE Port Health           |No Errors                 |In operating range        |
Fru Health               |In operating range        |In operating range        |
Security Violations      |No Errors                 |No Errors                 |
Fabric State Changes     |No Errors                 |Out of operating range    |
Switch Resource          |In operating range        |In operating range        |
Fabric Performance Impact|In operating range        |In operating range        |
IO Latency               |In operating range        |In operating range        |


3.2 Rules Affecting Health:
===========================

Category(Violation Count)|Repeat|Rule Name                  |Execution Time    |Object |Triggered   |
                         |Count |                           |                  |       |Value(Units)|
-----------------------------------------------------------------------------------------------------
Port Health(10)          |2     |defALL_16GSWL_SFPCURRENT_2 |05/18/20 22:36:00 |SFP 1  |0 mAmps     |
```

```
                        |      |                            |         |SFP 1           |0 mAmps     |
                        |4     |defALL_32GSWL_SFPTXP_251    |05/18/20 22:36:00 |SFP 22 |0 uW        |
                        |      |                            |         |SFP 17          |0 uW        |
                        |      |                            |         |SFP 22          |0 uW        |
                        |      |                            |         |SFP 17          |0 uW        |
                        |      |                            |         |SFP 22          |0 uW        |
                        |4     |defALL_32GSWL_SFPCURRENT_1  |05/18/20 22:36:00 |SFP 22 |0 mAmps     |
                        |      |                            |         |SFP 17          |0 mAmps     |
                        |      |                            |         |SFP 22          |0 mAmps     |
                        |      |                            |         |SFP 17          |0 mAmps     |
                        |      |                            |         |SFP 22          |0 mAmps     |
 Fabric State Changes(5)  |5     |defSWITCHFLOGI_4            |05/18/20 22:31:30 |Switch |11 Logins   |
                        |      |                            |         |Switch          |5 Logins    |
                        |      |                            |         |Switch          |10 Logins   |
                        |      |                            |         |Switch          |5 Logins    |
                        |      |                            |         |Switch          |5 Logins    |
```

Viewing MAPS Dashboard Settings

The following example uses the GET request to display the date and time when monitoring began with MAPS.

Structure

```
GET <base_URI>/running/brocade-maps/dashboard-misc
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/dashboard-misc
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <dashboard-misc>
        <maps-start-time>Wed Jun 27 07:35:06 2018</maps-start-time>
    </dashboard-misc>
</Response>
```

Clearing the MAPS Dashboard Data

The following example uses the PATCH request to clear the dashboard data for MAPS.

Structure

```
PATCH <base_URI>/running/brocade-maps/dashboard-misc/clear-data/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-maps/dashboard-misc/clear-data/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing MAPS Rules

The following example uses the GET request to display the configured MAPS rules. To display a specific MAPS rule, specify the rule name in the request.

For example, the example shows specifying the rule name (defALL_100G_QSFPCURRENT_10) in the request:

```
GET https://10.10.10.10/rest/running/brocade-maps/rule/name/defALL_100G_QSFPCURRENT_10
```

Structure

```
GET <base_URI>/running/brocade-maps/rule
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/rule
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <rule>
        <name>defALL_100G_QSFPCURRENT_10</name>
        <is-rule-on-rule>false</is-rule-on-rule>
        <monitoring-system>CURRENT</monitoring-system>
        <time-base>NONE</time-base>
        <logical-operator>ge</logical-operator>
        <threshold-value>10</threshold-value>
        <group-name>ALL_FCOE_100G_SR4_QSFP</group-name>
        <actions>
            <action>raslog</action>
            <action>snmp-trap</action>
            <action>e-mail</action>
            <action>sfp-marginal</action>
        </actions>
        <is-predefined>true</is-predefined>
    </rule>
...
    <rule>
        <name>ALL_PORTS_IO_FRAME_LOSS_UNQUARtest</name>
        <is-rule-on-rule>false</is-rule-on-rule>
        <monitoring-system>DEV_LATENCY_IMPACT</monitoring-system>
        <time-base>NONE</time-base>
```

```
        <logical-operator>eq</logical-operator>
        <threshold-value>IO_FRAME_LOSS</threshold-value>
        <group-name>ALL_PORTS</group-name>
        <actions>
            <action>raslog</action>
            <action>snmp-trap</action>
            <action>e-mail</action>
            <action>port-toggle</action>
            <action>un-quarantine</action>
            <action>sddq</action>
        </actions>
        <is-predefined>false</is-predefined>
        <toggle-time>2</toggle-time>
        <quiet-time>60</quiet-time>
        <un-quarantine-timeout>1</un-quarantine-timeout>
    </rule>
</Response>
```

Configuring a MAPS Rule

The following example uses the POST request to create new MAPS rule BaseRule_statechanges12.

Structure

```
POST <base_URI>/running/brocade-maps/rule
```

URI

```
POST https://10.10.10.10/rest/running/brocade-maps/rule
```

Request Body

```
    <rule>
        <name>BaseRule_statechanges12</name>
        <is-rule-on-rule>false</is-rule-on-rule>
        <monitoring-system>STATE_CHG</monitoring-system>
        <time-base>min</time-base>
        <logical-operator>g</logical-operator>
        <threshold-value>1</threshold-value>
        <group-name>ALL_F_PORTS</group-name>
        <actions>
            <action>raslog</action>
            <action>snmp-trap</action>
            <action>e-mail</action>
        </actions>
        <quiet-time>3900</quiet-time>
        <event-severity>WARNING</event-severity>
    </rule>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Manually Clearing Quiet Time

The following example uses the PATCH request to manually clear quiet time for RASlog and e-mail alerts.

Structure

```
PATCH <base_URI>/running/brocade-maps/rule/name/rule_name/quiet-time/0
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-maps/rule/name/BaseRule_statechanges12/quiet-time/0
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Manually Clearing Quarantined Ports

The following example uses the PATCH request to manually clear quarantined ports.

Structure

```
PATCH <base_URI>/running/brocade-maps/rule/name/rule_name/un-quarantine-clear/true|false
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-maps/rule/name/BaseRule_statechanges12/un-quarantine-clear/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring a MAPS Policy

The following example uses the POST request to create a new MAPS policy test_aggressive_policy. You can use a PATCH request to edit an existing policy.

Structure

```
POST <base_URI>/running/brocade-maps/maps-policy
```

URI

```
POST https://10.10.10.10/rest/running/brocade-maps/maps-policy
```

Request Body

```
<maps-policy>
    <name>test_aggressive_policy</name>
    <rule-list>
        <rule>defALL_100G_QSFPCURRENT_10</rule>
```

```
        <rule>defALL_100G_QSFPCURRENT_2</rule>
        <rule>defALL_100G_QSFPRXP_2187</rule>
        <rule>defSWITCHZONE_CHG_2</rule>
    </rule-list>
</maps-policy>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Activating a MAPS Policy

The following example uses the PATCH request to activate the MAPS policy test_aggressive_policy.

Structure

```
PATCH <base_URI>/running/brocade-maps/maps-policy/name/policy_name/is-active-policy/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-maps/maps-policy/name/test_aggressive_policy/is-active-policy/
true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the Credit Stall Dashboard

This example uses a GET request to display the Credit Stall dashboard.

Structure

```
GET base_URI/running/brocade-maps/credit-stall-dashboard
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/credit-stall-dashboard
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <credit-stall-dashboard>
        <slot-port>5/0</slot-port>
        <timestamp>2020-02-26T15:20:00-07:00</timestamp>
        <state>no-congestion</state>
        <frequency>0</frequency>
    </credit-stall-dashboard>
    <credit-stall-dashboard>
```

```
        <slot-port>5/0</slot-port>
        <timestamp>2020-02-26T15:19:00-07:00</timestamp>
        <state>no-congestion</state>
        <frequency>0</frequency>
    </credit-stall-dashboard>
...
    <credit-stall-dashboard>
        <slot-port>8/63</slot-port>
        <timestamp>2020-02-26T11:56:00-07:00</timestamp>
        <state>no-congestion</state>
        <frequency>0</frequency>
    </credit-stall-dashboard>
    <credit-stall-dashboard>
        <slot-port>8/63</slot-port>
        <timestamp>2020-02-26T11:55:00-07:00</timestamp>
        <state>no-congestion</state>
        <frequency>0</frequency>
    </credit-stall-dashboard>
</Response>
```

Viewing the Oversubscription Dashboard

This example uses a GET request to display the Oversubscription dashboard.

Structure

```
GET base_URI/running/brocade-maps/oversubscription-dashboard
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/oversubscription-dashboard
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <oversubscription-dashboard>
        <slot-port>5/0</slot-port>
        <timestamp>2020-02-26T15:24:00-07:00</timestamp>
        <state>no-oversubscription</state>
    </oversubscription-dashboard>
    <oversubscription-dashboard>
        <slot-port>5/0</slot-port>
        <timestamp>2020-02-26T15:23:00-07:00</timestamp>
        <state>no-oversubscription</state>
    </oversubscription-dashboard>
...
    <oversubscription-dashboard>
        <slot-port>8/63</slot-port>
        <timestamp>2020-02-26T11:56:00-07:00</timestamp>
```

```
        <state>no-oversubscription</state>
    </oversubscription-dashboard>
    <oversubscription-dashboard>
        <slot-port>8/63</slot-port>
        <timestamp>2020-02-26T11:55:00-07:00</timestamp>
        <state>no-oversubscription</state>
    </oversubscription-dashboard>
</Response>
```

Viewing the Dashboard History

This example uses a GET request to display the dashboard history.

Structure

```
GET base_URI/running/brocade-maps/dashboard-history
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/dashboard-history
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <dashboard-history>
        <category>fc-port</category>
        <date>current</date>
        <crc-errors/>
        <itw-errors/>
        <port-loss/>
        <loss-of-sync/>
        <link-failure>
            <port-data>8/56:7</port-data>
            <port-data>7/38:1</port-data>
            <port-data>7/35:1</port-data>
        </link-failure>
        <signal-loss>
            <port-data>7/4:4</port-data>
            <port-data>7/35:1</port-data>
            <port-data>7/38:1</port-data>
        </signal-loss>
        <protocol-errors/>
        <state-change>
            <port-data>8/56:14</port-data>
            <port-data>8/57:14</port-data>
            ...
            <port-data>8/58:1</port-data>
            <port-data>8/59:1</port-data>
        </state-change>
        <link-reset>
```

```
            <port-data>8/56:17</port-data>
            <port-data>8/57:9</port-data>
            ...
            <port-data>8/59:2</port-data>
            <port-data>7/6:1</port-data>
        </link-reset>
        <c3-tx-to/>
        <rx-perf>
            <port-data>7/42:4.49</port-data>
            <port-data>8/59:2.99</port-data>
            <port-data>8/58:1.14</port-data>
        </rx-perf>
        <tx-perf>
            <port-data>7/42:4.51</port-data>
            <port-data>8/59:3.71</port-data>
            <port-data>8/58:1.13</port-data>
        </tx-perf>
        <utilization>
            <port-data>7/42:4.50</port-data>
            <port-data>8/59:3.35</port-data>
            <port-data>8/58:1.13</port-data>
        </utilization>
    </dashboard-history>
    ...
 </Response>
```

Configuring an FPI Profile

This example uses a PATCH request to configure an FPI profile.

Structure

```
PATCH base_URI/running/brocade-maps/fpi-profile/name/[name]
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-maps/fpi-profile/name/new_fpi_profile
```

Request Body

```
<fpi-profile>
    <name>new_fpi_profile</name>
    <performance-impact>
        <transmit-queue-latency>10</transmit-queue-latency>
        <credit-zero-percentage-1-sec>70</credit-zero-percentage-1-sec>
        <credit-zero-percentage-5-sec>50</credit-zero-percentage-5-sec>
        <credit-zero-percentage-10-sec>30</credit-zero-percentage-10-sec>
    </performance-impact>
    <frame-loss>
        <transmit-queue-latency>80</transmit-queue-latency>
    </frame-loss>
 </fpi-profile>
```

Response Body

When the resource is successfully deleted, the response contains an empty message body and a "204 No Content" status in the first line of the headers.

Viewing the MAPS Device Login Data

This example uses a GET request to display the ports connected to FI as one group or trunk.

Structure

```
GET base_URI/running/brocade-maps/maps-device-login
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/maps-device-login
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <maps-device-login>
        <node-wwn>10:11:00:de:fb:a1:17:01</node-wwn>
        <state>BALANCED</state>
        <master-port>7/6</master-port>
        <ports>
            <port>7/6(3)</port>
            <port>7/7(3)</port>
        </ports>
    </maps-device-login>
    <maps-device-login>
        <node-wwn>10:1d:00:de:fb:a2:17:01</node-wwn>
        <state>BALANCED</state>
        <master-port>7/26</master-port>
        <ports>
            <port>7/26(3)</port>
            <port>7/27(3)</port>
        </ports>
    </maps-device-login>
</Response>
```

Viewing the Backend Port History

This example uses a GET request to display the backend port history.

Structure

```
GET base_URI/running/brocade-maps/backend-ports-history
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/backend-ports-history
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
<Response>
    <backend-ports-history>
        <category>be-port</category>
        <date>03/31/2022</date>
        <crc-errors>
            <port-data>8/56:7</port-data>
            <port-data>7/38:1</port-data>
            <port-data>7/35:1</port-data>
        </crc-errors>
        <itw-errors>
            <port-data>8/56:14</port-data>
            <port-data>8/57:14</port-data>
        </itw-errors>
        <link-reset>
            <port-data>8/56:7</port-data>
            <port-data>7/38:1</port-data>
            <port-data>7/35:1</port-data>
        </link-reset>
        <bad-os></>
           <port-data>8/56:17</port-data>
            <port-data>8/57:9</port-data>
        </bad-os>
        <long-frame>
            <port-data>8/56:17</port-data>
            <port-data>8/57:9</port-data>
        </long-frame>
        <truncated-frame>
            <port-data>7/42:4.50</port-data>
            <port-data>8/59:3.35</port-data>
            <port-data>8/58:1.13</port-data>
        </truncated-frame>
    </backend-ports-history>
</Response>
```

Viewing the Gigabit Ethernet Ports History

This example uses a GET request to display the GE port history.

Structure

```
GET base_URI/running/brocade-maps/gigabit-ethernet-ports-history
```

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/gigabit-ethernet-ports-history
```

Request Body

No request body is required.

Response Body

```
<?xml version="1.0"?>
```

```
<Response>
    <gigabit-ethernet-ports-history>
        <category>ge-port</category>
        <date>03/31/2022</date>
        <ge-crc-errors>
            <port-data>8/56:7</port-data>
            <port-data>7/38:1</port-data>
            <port-data>7/35:1</port-data>
        </ge-crc-errors>
        <ge-signal-loss>
            <port-data>8/56:14</port-data>
            <port-data>8/57:14</port-data>
        </ge-signal-loss>
    </gigabit-ethernet-ports-history>
</Response>
```

brocade-media

This module provides a detailed summary of the Small Form-factor Pluggable (SFP) transceivers media data for all available ports. For each SFP, the summary includes information that describes the SFP capabilities, interfaces, manufacturer, and other information. Refer to the module tree below for a complete list of parameters included in the SFP summary.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-media.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Typesfor data type descriptions.

```
module: brocade-media
  +--ro brocade-media
     +--ro media-rdp* [name]
        +--ro name                                interface-type
        +--ro identifier?                         string
        +--ro connector?                          string
        +--ro nominal-baud-rate?                  int32
        +--ro baud-rate-maximum?                  int32
        +--ro baud-rate-minimum?                  int32
        +--ro optical-cable-types
        |  +--ro cable-type*   cable-descriptor-type
        +--ro transmission-type?                  transmission-types
        +--ro fiber-9um-link-length?              uint32
        +--ro fiber-9m-link-length?               uint32
        +--ro multimode-fiber-om2-50um-length?    uint32
        +--ro multimode-fiber-om3-50um-length?    uint32
        +--ro multimode-fiber-om4-50um-length?    uint32
        +--ro multimode-fiber-625um-length?       uint32
        +--ro copper-link-length?                 uint32
        +--ro enhanced-options?                   fibrechannel:brocade-hex-string-type
        +--ro digital-diagnostics-type?           fibrechannel:brocade-hex-string-type
        +--ro media-optional-signals
        |  +--ro signal*   signal-descriptor-type
        +--ro media-optional-signal-value?        string
        +--ro status-control?                     fibrechannel:brocade-hex-string-type
        +--ro electric-loopback-enabled?          boolean
        +--ro optical-loopback-enabled?           boolean
        +--ro media-speed-capability
        |  +--ro speed*   capable-speed-type
        +--ro media-distance
        |  +--ro distance*   string
        +--ro encoding?                           string
        +--ro vendor-oui?                         string
        +--ro part-number?                        fru:part-number-type
        +--ro serial-number?                      fru:serial-number-type
        +--ro vendor-name?                        vendor-name-type
        +--ro vendor-revision?                    vendor-revision-type
        +--ro date-code?                          date-code-type
        +--ro temperature?                        temperature-type
        +--ro rx-power?                           media-power-type
```

```
+--ro tx-power?                              media-power-type
+--ro current?                               current-type
+--ro voltage?                               voltage-type
+--ro wavelength?                            uint32
+--ro power-on-time?                         int32
+--ro alarm-flags
|  +--ro high-alarm?   fibrechannel:brocade-hex-string-type
|  +--ro low-alarm?    fibrechannel:brocade-hex-string-type
+--ro warning-flags
|  +--ro high-warning?   fibrechannel:brocade-hex-string-type
|  +--ro low-warning?    fibrechannel:brocade-hex-string-type
+--ro state-transitions?                     uint32
+--ro last-poll-time?                        union
+--ro peer-data-available?                   boolean
+--ro remote-identifier?                     string
+--ro remote-laser-type?                     string
+--ro remote-optical-type?                   optical-type
+--ro remote-media-speed-capability
|  +--ro speed*   capable-speed-type
+--ro remote-optical-product-data
|  +--ro part-number?       fru:part-number-type
|  +--ro serial-number?     fru:serial-number-type
|  +--ro vendor-name?       vendor-name-type
|  +--ro vendor-revision?   vendor-revision-type
|  +--ro date-code?         date-code-type
+--ro remote-media-temperature?              temperature-type
+--ro remote-media-rx-power?                 media-power-type
+--ro remote-media-tx-power?                 media-power-type
+--ro remote-media-current?                  current-type
+--ro remote-media-voltage?                  voltage-type
o--ro remote-media-voltage-alert
|  o--ro high-alarm?     alert-type
|  o--ro low-alarm?      alert-type
|  o--ro high-warning?   alert-type
|  o--ro low-warning?    alert-type
+--ro remote-media-signal-loss
|  +--ro up-stream?     stream-type
|  +--ro down-stream?   stream-type
o--ro remote-media-temperature-alert
|  o--ro high-alarm?     alert-type
|  o--ro low-alarm?      alert-type
|  o--ro high-warning?   alert-type
|  o--ro low-warning?    alert-type
o--ro remote-media-tx-bias-alert
|  o--ro high-alarm?     alert-type
|  o--ro low-alarm?      alert-type
|  o--ro high-warning?   alert-type
|  o--ro low-warning?    alert-type
o--ro remote-media-tx-power-alert
|  o--ro high-alarm?     alert-type
|  o--ro low-alarm?      alert-type
|  o--ro high-warning?   alert-type
|  o--ro low-warning?    alert-type
```

```
o--ro remote-media-rx-power-alert
|  o--ro high-alarm?     alert-type
|  o--ro low-alarm?      alert-type
|  o--ro high-warning?   alert-type
|  o--ro low-warning?    alert-type
+--ro remote-media-voltage-alarm-type
|  +--ro high-alarm?     alarm-type
|  +--ro low-alarm?      alarm-type
|  +--ro high-warning?   alarm-type
|  +--ro low-warning?    alarm-type
+--ro remote-media-temperature-alarm-type
|  +--ro high-alarm?     temp-alarm-type
|  +--ro low-alarm?      temp-alarm-type
|  +--ro high-warning?   temp-alarm-type
|  +--ro low-warning?    temp-alarm-type
+--ro remote-media-tx-bias-alarm-type
|  +--ro high-alarm?     alarm-type
|  +--ro low-alarm?      alarm-type
|  +--ro high-warning?   alarm-type
|  +--ro low-warning?    alarm-type
+--ro remote-media-tx-power-alarm-type
|  +--ro high-alarm?     alarm-type
|  +--ro low-alarm?      alarm-type
|  +--ro high-warning?   alarm-type
|  +--ro low-warning?    alarm-type
+--ro remote-media-rx-power-alarm-type
   +--ro high-alarm?     alarm-type
   +--ro low-alarm?      alarm-type
   +--ro high-warning?   alarm-type
   +--ro low-warning?    alarm-type
```

Examples

This section provides an example for using the brocade-media module.

Viewing the Temperature Alerts for a Specific Interface

The following example uses the GET request to view the SFP temperature alerts for a specific interface (fc/7/41).

Structure

```
GET <base_URI>/running/brocade-media/media-rdp/name/interface-id/slot/port/remote-media-temperature-alert
```

URI

```
GET https://10.10.10.10/rest/running/brocade-media/media-rdp/name/fc%2f7%2f41/remote-media-temperature-alert
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <media-rdp>
        <name>fc/7/41</name>
        <remote-media-temperature-alert>
            <high-alarm>85</high-alarm>
            <low-alarm>-5</low-alarm>
            <high-warning>75</high-warning>
            <low-warning>0</low-warning>
        </remote-media-temperature-alert>
    </media-rdp>
</Response>
```

brocade-name-server

This module is used to monitor the operation of one or more instances of Name Server functionality.

For a list of changes to this module by release, refer to the individual module YANG file.

NOTE
The brocade-name-server module is supported in Fabric OS 8.2.0a and later.

The brocade-name-server module supports the following query parameter combinations:

- port-id
- port-name
- node-name

Module Tree

This is the tree view of the module from the `brocade-name-server.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-name-server
  +--ro brocade-name-server
     +--ro fibrechannel-name-server* [port-id] {fibrechannel:fibrechannel_switch_platform}?
        +--ro port-id                          fibrechannel:fcid-hex-string-type
        +--ro port-name?                       fibrechannel:wwn-type
        +--ro port-symbolic-name?              string
        +--ro fabric-port-name?                fibrechannel:wwn-type
        +--ro permanent-port-name?             fibrechannel:wwn-type
        +--ro node-name?                       fibrechannel:wwn-type
        +--ro node-symbolic-name?              string
        +--ro class-of-service?                fibrechannel:class-of-service-type
        +--ro fc4-type?                        fibrechannel:fc4-type-type
        +--ro fc4-features?                    fibrechannel:fc4-features-type
        +--ro port-type?                       fibrechannel:port-type-string-type
        +--ro state-change-registration?       string
        +--ro name-server-device-type?         string
        +--ro port-index?                      fibrechannel:user-port-number-type
        o--ro share-area?                      string
        +--ro frame-redirection?               string
        +--ro partial?                         string
        +--ro lsan?                            enumeration
        x--ro link-speed?                      string
        +--ro protocol-speed?                  fibrechannel:protocol-speed-type
        +--ro port-properties?                 string
        +--ro cascaded-ag?                     string
        +--ro connected-through-ag?            string
        +--ro real-device-behind-ag?           string
        +--ro fcoe-device?                     string
        +--ro slow-drain-device-quarantine?    string
        +--ro connected-through-fc-lag?        boolean
```

Examples

This section provides a few basic examples for using the brocade-name-server module.

Retrieving Name Server Information for a Switch

This example uses a GET request to retrieve the Name Server information for a switch.

Structure

```
GET <base_URI>/running/brocade-name-server/fibrechannel-name-server
```

URI Request

```
GET https://10.10.10.10/rest/running/brocade-name-server/fibrechannel-name-server
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-name-server>
        <port-id>0x2b9200</port-id>
        <port-name>10:00:00:10:9b:8f:2b:c7</port-name>
        <port-symbolic-name>[34] &quot;Emulex PPN-10:00:00:10:9B:8F:2B:C7&quot;</port-symbolic-name>
        <fabric-port-name>20:92:c4:f5:7c:4a:ac:6c</fabric-port-name>
        <permanent-port-name>10:00:00:10:9b:8f:2b:c7</permanent-port-name>
        <node-name>20:00:00:10:9b:8f:2b:c7</node-name>
        <node-symbolic-name>[76] &quot;Emulex LPe35002-M2 FV12.8.351.6901 DV12.6.165.0 HN:W16-68166 OS:Windows
 2016&quot;</node-symbolic-name>
        <class-of-service>class-2 class-3</class-of-service>
        <fc4-type>FCP</fc4-type>
        <fc4-features>FCP-Initiator</fc4-features>
        <port-type>n-port</port-type>
        <state-change-registration>Fabric-Detected Nx-Port-Detected</state-change-registration>
        <name-server-device-type>Physical Initiator</name-server-device-type>
        <port-index>146</port-index>
        <frame-redirection>no</frame-redirection>
        <partial>no</partial>
        <lsan>no</lsan>
        <link-speed>32G</link-speed>
        <protocol-speed>32-gfc</protocol-speed>
        <port-properties></port-properties>
        <cascaded-ag>no</cascaded-ag>
        <connected-through-ag>no</connected-through-ag>
        <real-device-behind-ag>no</real-device-behind-ag>
        <fcoe-device>no</fcoe-device>
        <slow-drain-device-quarantine>no</slow-drain-device-quarantine>
        <connected-through-fc-lag>false</connected-through-fc-lag>
    </fibrechannel-name-server>
     <fibrechannel-name-server>
        <port-id>0xee2a00</port-id>
        <port-name>20:28:d8:1f:cc:5a:5d:e8</port-name>
        <port-symbolic-name></port-symbolic-name>
        <fabric-port-name>20:2a:d8:1f:cc:5a:f8:4e</fabric-port-name>
```

```
        <permanent-port-name>20:28:d8:1f:cc:5a:5d:e8</permanent-port-name>
        <node-name>10:00:d8:1f:cc:5a:5d:e8</node-name>
        <node-symbolic-name>[78] &quot;Brocade-AG | 66-63_G720_Cassian | 181 |
 2620:100:4:fa01:da1f:ccff:fe5a:5de8 | &quot;</node-symbolic-name>
        <class-of-service>class-3</class-of-service>
        <fc4-type></fc4-type>
        <fc4-features></fc4-features>
        <port-type>n-port</port-type>
        <state-change-registration></state-change-registration>
        <name-server-device-type>Physical Unknown(initiator/target)</name-server-device-type>
        <port-index>42</port-index>
        <frame-redirection>no</frame-redirection>
        <partial>no</partial>
        <lsan>unknown</lsan>
        <link-speed>64G</link-speed>
        <protocol-speed>64-gfc</protocol-speed>
        <port-properties></port-properties>
        <cascaded-ag>no</cascaded-ag>
        <connected-through-ag>yes</connected-through-ag>
        <real-device-behind-ag>no</real-device-behind-ag>
        <fcoe-device>no</fcoe-device>
        <slow-drain-device-quarantine>no</slow-drain-device-quarantine>
        <connected-through-fc-lag>false</connected-through-fc-lag>
    </fibrechannel-name-server>
</Response>
```

Retrieving Name Server information for a Port

This example uses a GET request to retrieve the Name Server information for a specific port.

Structure

```
GET <base_URI>/running/brocade-name-server/fibrechannel-name-server/port-id
```

URI Request

```
GET https://10.10.10.10/running/brocade-name-server/fibrechannel-name-server/port-id/0x2b9200
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-name-server>
        <port-id>0x2b9200</port-id>
        <port-name>10:00:00:10:9b:8f:2b:c7</port-name>
        <port-symbolic-name>[34] "Emulex PPN-10:00:00:10:9B:8F:2B:C7"</port-symbolic-name>
        <fabric-port-name>20:92:c4:f5:7c:4a:ac:6c</fabric-port-name>
        <permanent-port-name>10:00:00:10:9b:8f:2b:c7</permanent-port-name>
```

```
        <node-name>20:00:00:10:9b:8f:2b:c7</node-name>
        <node-symbolic-name>[76] "Emulex LPe35002-M2 FV12.8.351.6901 DV12.6.165.0 HN:W16-68166 OS:Windows
 2016"</node-symbolic-name>
        <class-of-service>class-2 class-3</class-of-service>
        <fc4-type>FCP</fc4-type>
        <fc4-features>FCP-Initiator</fc4-features>
        <port-type>n-port</port-type>
        <state-change-registration>Fabric-Detected Nx-Port-Detected</state-change-registration>
        <name-server-device-type>Physical Initiator</name-server-device-type>
        <port-index>146</port-index>
        <frame-redirection>no</frame-redirection>
        <partial>no</partial>
        <lsan>no</lsan>
        <link-speed>32G</link-speed>
        <protocol-speed>32-gfc</protocol-speed>
        <port-properties/>
        <cascaded-ag>no</cascaded-ag>
        <connected-through-ag>no</connected-through-ag>
        <real-device-behind-ag>no</real-device-behind-ag>
        <fcoe-device>no</fcoe-device>
        <slow-drain-device-quarantine>no</slow-drain-device-quarantine>
        <connected-through-fc-lag>false</connected-through-fc-lag>
    </fibrechannel-name-server>
</Response>
```

brocade-security

This module provides a detailed view of Fabric OS System Security configuration.

The brocade-security module enables you to manage the following features:

- IP filter policies
- IP filter rules
- User accounts
- Password configuration
- SSH configuration
- Remote authentication
- Security protocols
- Security certificate
- AAA services
- LDAP role mapping

It assumes a knowledge of System Security as performed in Fabric OS. For information on these topics, refer to the *Brocade Fabric OS Administration Guide*.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-security.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-security
  +--rw brocade-security
     +--rw ipfilter-policy* [name]
     |  +--rw name                              brocade-security-type:ipfilter-name-type
     |  +--rw ip-version?                       brocade-security-type:ipfilter-ip-version-type
     |  +--ro is-policy-active?                 boolean
     |  +--ro is-default-policy?                boolean
     |  +--rw action?                           brocade-security-type:ipfilter-action-type
     |  +--rw clone-destination-policy-name?    brocade-security-type:ipfilter-name-type
     +--rw ipfilter-rule* [policy-name index]
     |  +--rw policy-name                 brocade-security-type:ipfilter-name-type
     |  +--rw index                       uint16
     |  +--rw source-ip?                  brocade-security-type:host-type
     |  +--rw destination-start-port      inet:port-number
     |  +--rw destination-end-port?       inet:port-number
     |  +--rw protocol?                   brocade-security-type:ipfilter-protocol-type
     |  +--rw permission                  brocade-security-type:ipfilter-permission-type
     |  o--rw traffic-type?               brocade-security-type:ipfilter-traffic-type
     |  +--rw destination-ip?             brocade-security-type:host-type
     +--rw user-specific-password-cfg* [user-name]
     |  +--rw user-name                brocade-security-type:user-config-user-name-type
     |  +--rw minimum-password-age?    int32
     |  +--rw maximum-password-age?    int32
     |  +--rw warn-on-expire?          int32
```

```
|  +--rw enforce-expire?         boolean
|  +--ro hash-type?              brocade-security-type:hash-algorithm-type
+--rw password-cfg
|  +--rw minimum-length?                          int32
|  +--rw character-set?                           int32
|  +--rw user-name-allowed?                       boolean
|  +--rw minimum-lower-case-character?            int32
|  +--rw minimum-upper-case-character?            int32
|  +--rw minimum-numeric-character?               int32
|  +--rw minimum-special-character?               int32
|  +--rw past-password-history?                   int32
|  +--rw minimum-password-age?                    int32
|  +--rw maximum-password-age?                    int32
|  +--rw warn-on-expire?                          int32
|  +--rw lock-out-threshold?                      int32
|  +--rw lock-out-duration?                       int32
|  +--rw admin-lock-out-enabled?                  boolean
|  +--rw admin-lock-out-console-access-enabled?   boolean
|  +--rw repeat-character-limit?                  int32
|  +--rw sequence-character-limit?                int32
|  +--ro password-config-changed?                 boolean
|  +--rw reverse-user-name-allowed?               boolean
|  +--rw hash-type?                               brocade-security-type:hash-algorithm-type
|  +--rw manual-hash-enabled?                     boolean
|  +--rw enforce-expire?                          boolean
|  +--rw minimum-difference?                      uint16
|  +--rw password-action?                         brocade-security-type:password-cfg-operation-type
+--rw user-config* [name]
|  +--rw name                           brocade-security-type:user-config-login-user-name-type
|  +--rw password?                      brocade-security-type:user-password-type
|  +--rw role?                          brocade-security-type:user-config-role-type
|  +--rw account-description?           string
|  +--rw account-enabled?               boolean
|  +--ro default-password-configured?   boolean
|  +--rw password-change-enforced?      boolean
|  +--rw account-locked?                boolean
|  +--rw access-start-time?             string
|  +--rw access-end-time?               string
|  +--rw home-virtual-fabric?           home-virtual-fabric-type
|  +--rw chassis-access-role?           user-config-role-type
|  +--rw virtual-fabric-role-id-list
|  |  +--rw role-id*   brocade-security-type:virtual-fabric-role-id-type
|  +--ro auth-token-present?            boolean
+--rw role-config* [name]
|  +--rw name                 brocade-security-type:role-name-type
|  +--rw role-description?    string
|  +--ro role-type?           brocade-security-type:rbac-role-type
|  +--rw class-permissions
|     +--rw class-permission*   brocade-security-type:rbac-class-permission-type
+--rw auth-spec
|  +--rw authentication-mode?         brocade-security-type:aaa-authspec-type
|  +--rw activate-no-log-out?         boolean
|  +--rw primary-auth-log-messages?   boolean
```

```
+--rw radius-server* [server]
|  +--rw server             inet:host
|  +--rw port?              inet:port-number
|  +--rw secret?            string
|  +--rw timeout?           brocade-security-type:aaa-timeout-type
|  +--rw authentication?    union
|  +--rw encryption-type?   brocade-security-type:aaa-encryption-algorithm-type
|  +--rw position?          uint16
+--rw tacacs-server* [server]
|  +--rw server             inet:host
|  +--rw port?              inet:port-number
|  +--rw secret?            string
|  +--rw timeout?           brocade-security-type:aaa-timeout-type
|  +--rw authentication?    brocade-security-type:aaa-protocols-type
|  +--rw encryption-type?   brocade-security-type:aaa-encryption-algorithm-type
|  +--rw position?          uint16
+--rw ldap-server* [server]
|  +--rw server      inet:host
|  +--rw port?       inet:port-number
|  +--rw domain?     inet:domain-name
|  +--rw timeout?    brocade-security-type:aaa-timeout-type
|  +--rw tls-mode?   brocade-security-type:ldap-tls-mode-type
|  +--rw position?   uint16
+--rw ldap-role-map* [ldap-role]
|  +--rw ldap-role              string
|  +--rw switch-role?           string
|  +--rw home-virtual-fabric?   home-virtual-fabric-type
|  +--rw chassis-access-role?   user-config-role-type
+--rw sec-crypto-cfg-template-action
|  +--rw template-name?                brocade-security-type:sec-crypto-cfg-template-name-type
|  +--rw action?                       brocade-security-type:sec-crypto-cfg-actions-type
|  +--rw remote-user-name?             string
|  +--rw remote-host-ip?               inet:host
|  +--rw remote-user-password?         string
|  +--rw remote-directory?             string
|  +--rw file-transfer-protocol-type?   sec-crypto-cfg-file-transfer-protocol-type
+--ro sec-crypto-cfg-template* [name]
|  +--ro name        brocade-security-type:sec-crypto-cfg-template-name-type
|  +--ro template?   string
+--rw sec-crypto-cfg
|  +--rw ssh-cipher?                 brocade-security-type:default-string-type
|  +--rw ssh-kex?                    brocade-security-type:default-string-type
|  +--rw ssh-mac?                    brocade-security-type:default-string-type
|  +--rw https-cipher?               brocade-security-type:default-string-type
|  +--rw https-tlsv13-cipher?        brocade-security-type:default-string-type
|  +--rw radius-cipher?              brocade-security-type:default-string-type
|  +--rw ldap-cipher?                brocade-security-type:default-string-type
|  +--rw syslog-cipher?              brocade-security-type:default-string-type
|  +--rw https-tls-protocol?         brocade-security-type:tls-protocol-type
|  +--rw radius-tls-protocol?        brocade-security-type:tls-protocol-type
|  +--rw ldap-tls-protocol?          brocade-security-type:tls-protocol-type
|  +--rw syslog-tls-protocol?        brocade-security-type:tls-protocol-type
|  +--rw x509v3-validation-mode?     brocade-security-type:validation-mode-type
```

```
|  +--ro crypto-version?             brocade-security-type:crypto-version-type
|  +--rw boot-up-self-test-enabled?  boolean
|  +--ro boot-up-self-test-status?   brocade-security-type:boot-up-self-test-status-type
+--rw sshutil
|  +--rw allow-user-name?   brocade-security-type:user-config-user-name-type
|  +--rw rekey-interval?    uint16
+--rw sshutil-known-host* [remote-host-name]
|  +--rw remote-host-name    brocade-security-type:known-host-type
|  +--ro key-type?           string
|  +--ro host-key?           string
|  +--rw fingerprint-list
|     +--rw fingerprint*   string
+--rw sshutil-key* [algorithm-type key-type]
|  +--rw algorithm-type    brocade-security-type:sshutil-algorithm-type
|  +--rw key-type          brocade-security-type:sshutil-key-type
|  o--rw passphrase?       string
|  +--ro size?             uint16
|  +--ro fingerprint?      string
|  +--rw key-size?         brocade-security-type:keysize-type
+--rw sshutil-public-key* [user-name]
|  +--rw user-name     brocade-security-type:user-config-user-name-type
|  +--ro public-key?   string
+--rw sshutil-public-key-action
|  +--rw action?                brocade-security-type:sshutil-operation-type
|  +--rw algorithm-type?        brocade-security-type:sshutil-algorithm-type
|  +--rw user-name?             brocade-security-type:user-config-user-name-type
|  +--rw remote-user-name?      string
|  +--rw remote-host-ip?        inet:host
|  +--rw remote-user-password?  string
|  +--rw remote-directory?      string
|  +--rw public-key-name?       string
+--rw password
|  +--rw user-name?      brocade-security-type:user-config-user-name-type
|  +--rw old-password?   brocade-security-type:user-password-type
|  +--rw new-password?   brocade-security-type:user-password-type
+--rw security-certificate-generate
|  +--rw certificate-entity?                      brocade-security-type:gen-certificate-entity-type
|  +--rw certificate-type?                        brocade-security-type:certificate-application-type
|  +--rw algorithm-type?                          brocade-security-type:sshutil-algorithm-type
|  +--rw key-size?                                brocade-security-type:keysize-type
|  +--rw hash-type?                               brocade-security-type:seccertmgmt-hash-type
|  +--rw years?                                   uint16
|  +--rw keypair-tag?                             string
|  +--rw country-name?                            string
|  +--rw state-name?                              string
|  +--rw locality-name?                           string
|  +--rw organization-name?                       string
|  o--rw unit-name?                               string
|  +--rw domain-name?                             string
|  +--rw email-address?                           string
|  o--rw ip-address-in-subject-alternative-name?  boolean
|  +--rw subject-alternative-name-dns-names
|  |  +--rw dns-name*   inet:host
```

```
|  +--rw subject-alternative-name-ip-addresses
|     +--rw ip-address*   inet:ip-address
+--rw security-certificate-action
|  +--rw remote-user-name?       string
|  +--rw remote-host-ip?         inet:host
|  +--rw remote-user-password?   string
|  +--rw remote-directory?       string
|  +--rw protocol?               brocade-security-type:seccertmgmt-protocol-type
|  +--rw certificate-entity?     brocade-security-type:certificate-entity-type
|  +--rw certificate-type?       brocade-security-type:certificate-application-type
|  +--rw certificate-name?       string
|  +--rw ca-certificate?         string
|  +--rw operation?              brocade-security-type:seccertmgmt-operation-type
|  +--rw keypair-tag?            string
+--ro security-certificate* [certificate-entity certificate-type]
|  +--ro certificate-entity     brocade-security-type:certificate-entity-type
|  +--ro certificate-type       union
|  +--ro certificate?           string
|  +--ro certificate-hexdump?   brocade-security-type:certificate-hexdump-type
|  +--ro certificate-verbose?   string
+--ro security-certificate-extension* [] {fibrechannel:fibrechannel_extension_platform}?
|  +--ro certificate-entity?    brocade-security-type:extension-certificate-entity-type
|  +--ro local-certificate?     boolean
|  +--ro certificate-name?      string
|  +--ro certificate?           string
|  +--ro keypair-tag?           string
|  +--ro certificate-hexdump?   brocade-security-type:certificate-hexdump-type
|  +--ro certificate-verbose?   string
+--ro management-rbac-map* []
|  +--ro url?            string
|  +--ro rbac-class?     brocade-security-type:rbac-class-type
|  +--ro context-type?   brocade-security-type:rbac-context-type
+--ro acl-policy* [] {fibrechannel:fibrechannel_switch_platform}?
|  +--ro policy-activated?   boolean
|  +--ro policy-name?        brocade-security-type:security-policy-name-type
+--ro defined-fcs-policy-member-list* [] {fibrechannel:fibrechannel_switch_platform}?
|  +--ro switch-wwn?       fibrechannel:wwn-type
|  +--ro domain-id?        fibrechannel:domain-id-type
|  +--ro switch-name?      string
|  +--ro is-fcs-primary?   boolean
|  +--ro fcs-position?     uint16
+--ro defined-scc-policy-member-list* [] {fibrechannel:fibrechannel_switch_platform}?
|  +--ro switch-wwn?    fibrechannel:wwn-type
|  +--ro domain-id?     fibrechannel:domain-id-type
|  +--ro switch-name?   string
+--ro defined-dcc-policy-member-list* [] {fibrechannel:fibrechannel_switch_platform}?
|  +--ro policy-name?   brocade-security-type:security-policy-name-type
|  +--ro wwn?           fibrechannel:wwn-type
|  +--ro member-type?   dcc-policy-member-type
|  +--ro domain-id?     fibrechannel:domain-id-type
|  +--ro switch-name?   string
|  +--ro ports
|     +--ro port*   fibrechannel:user-port-number-type
```

```
      +--ro active-fcs-policy-member-list* [] {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro switch-wwn?       fibrechannel:wwn-type
      |  +--ro domain-id?        fibrechannel:domain-id-type
      |  +--ro switch-name?      string
      |  +--ro is-fcs-primary?   boolean
      |  +--ro fcs-position?     uint16
      +--ro active-scc-policy-member-list* [] {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro switch-wwn?    fibrechannel:wwn-type
      |  +--ro domain-id?     fibrechannel:domain-id-type
      |  +--ro switch-name?   string
      +--ro active-dcc-policy-member-list* [] {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro policy-name?   brocade-security-type:security-policy-name-type
      |  +--ro wwn?           fibrechannel:wwn-type
      |  +--ro member-type?   dcc-policy-member-type
      |  +--ro domain-id?     fibrechannel:domain-id-type
      |  +--ro switch-name?   string
      |  +--ro ports
      |     +--ro port*   fibrechannel:user-port-number-type
      +--rw policy-distribution-config
      |  +--rw scc-distribute-permission?         brocade-security-type:security-policy-distribute-permission-
type {fibrechannel:fibrechannel_switch_platform}?
      |  +--rw dcc-distribute-permission?         brocade-security-type:security-policy-distribute-permission-
type {fibrechannel:fibrechannel_switch_platform}?
      |  +--rw password-distribute-permission?    brocade-security-type:security-policy-distribute-permission-
type
      |  +--rw fcs-distribute-permission?         brocade-security-type:security-policy-distribute-permission-
type {fibrechannel:fibrechannel_switch_platform}?
      |  +--rw auth-distribute-permission?        brocade-security-type:security-policy-distribute-permission-
type {fibrechannel:fibrechannel_switch_platform}?
      |  +--rw ip-filter-distribute-permission?   brocade-security-type:security-policy-distribute-permission-
type {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro fabric-wide-policies {fibrechannel:fibrechannel_switch_platform}?
      |     +--ro policy*   brocade-security-type:fabric-wide-consistency-policy-type
      +--ro security-policy-size {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro defined-size?   uint32
      |  +--ro active-size?    uint32
      |  +--ro max-size?       uint32
      +--ro security-violation-statistics* [domain-id] {fibrechannel:fibrechannel_switch_platform}?
      |  +--ro domain-id              fibrechannel:domain-id-type
      |  +--ro telnet-policy?         yang:zero-based-counter32
      |  +--ro http-policy?           yang:zero-based-counter32
      |  +--ro scc-policy?            yang:zero-based-counter32
      |  +--ro dcc-policy?            yang:zero-based-counter32
      |  +--ro login?                 yang:zero-based-counter32
      |  +--ro authentication-fail?   yang:zero-based-counter32
      |  +--ro ts-out-sync?           yang:zero-based-counter32
      |  +--ro illegal-cmd?           yang:zero-based-counter32
      +--ro dh-chap-authentication-secret* []
      |  +--ro switch-wwn?    fibrechannel:wwn-type
      |  +--ro domain-id?     fibrechannel:domain-id-type
      |  +--ro switch-name?   string
      |  +--ro device-wwn?    fibrechannel:wwn-type
      +--rw authentication-configuration
```

```
     |  +--rw authentication-type?           brocade-security-type:authentication-protocol-type
     |  +--rw authentication-hash-type?      brocade-security-type:authentication-hash-type
     |  +--rw dh-group?                      brocade-security-type:dh-group-type
     |  +--rw switch-authentication-policy?  brocade-security-type:authentication-policy-mode-type
     |  +--rw device-authentication-policy?  brocade-security-type:authentication-policy-mode-type
     +--ro rbac-class* []
        +--ro class-name?       brocade-security-type:rbac-class-type
        +--ro command-name?     string
        +--ro command-option?   string
        +--ro permission?       brocade-security-type:rbac-permission-type
        +--ro context?          brocade-security-type:rbac-context-type
{fibrechannel:fibrechannel_virtual_fabric_platform}?
```

brocade-security Examples

This section provides examples for the brocade-security module.

It assumes a knowledge of System Security as performed in Fabric OS. For information on these topics, refer to the *Brocade Fabric OS Administration Guide*.

Viewing all IP Filter Policies

The following example uses the GET request to display all IP filter policies configured on the device. You can view a specific rule in a specific IP filter policy configuration by specifying the IP filter policy and index number in the request (for example, GET *<base_URI>*/running/brocade-security/ipfilter-policy/fwd_ipv4/index/1).

Structure

GET *<base_URI>*/running/brocade-security/ipfilter-policy

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/ipfilter-policy
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <ipfilter-policy>
        <name>default_ipv4</name>
        <ip-version>IPv4</ip-version>
        <is-policy-active>true</is-policy-active>
        <is-default-policy>true</is-default-policy>
    </ipfilter-policy>
    <ipfilter-policy>
        <name>default_ipv6</name>
        <ip-version>IPv6</ip-version>
        <is-policy-active>true</is-policy-active>
        <is-default-policy>true</is-default-policy>
    </ipfilter-policy>
    <ipfilter-policy>
        <name>fwd_ipv4</name>
        <ip-version>IPv4</ip-version>
        <is-policy-active>false</is-policy-active>
        <is-default-policy>false</is-default-policy>
    </ipfilter-policy>
    <ipfilter-policy>
        <name>46RB7ATw0XsERUO2X5h</name>
        <ip-version>IPv4</ip-version>
        <is-policy-active>false</is-policy-active>
        <is-default-policy>false</is-default-policy>
    </ipfilter-policy>
</Response>
```

Configuring an IP Filter Policy

The following example uses the POST request to configure a new IP filter policy on the device. You can use a PATCH request to edit an existing IP policy filter. Note that there is no transactional state for an IP filter policy request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

POST *<base_URI>*/running/brocade-security/ipfilter-policy

URI Request

```
POST https://10.10.10.10/rest/running/brocade-security/ipfilter-policy
```

Request Body

```
<ipfilter-policy>
    <name>ipv4_new</name>
    <ip-version>IPv4</ip-version>
</ipfilter-policy>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Cloning an IP Filter Policy

The following example uses the PATCH request to clone IP filter policy test_policy with a new name destPolicy on the device. You can use a PATCH request to edit an existing IP policy filter. Note that there is no transactional state for an IP filter policy request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

PATCH *<base_URI>*/running/brocade-security/ipfilter-policy

URI Request

```
PATCH  https://10.10.10.10/rest/running/brocade-security/ipfilter-policy
```

Request Body

```
<ipfilter-policy>
      <name>test_policy</name>
      <action>clone</action>
      <clone-destination-policy-name>destPolicy</clone-destination-policy-name>
</ipfilter-policy>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Activating an IP Filter Policy

The following example uses the PATCH request to activate the destPolicy IP filter policy on the device. Note that there is no transactional state for an IP filter policy request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

PATCH *<base_URI>*/running/brocade-security/ipfilter-policy

URI Request

```
PATCH  https://10.10.10.10/rest/running/brocade-security/ipfilter-policy
```

Request Body

```
<ipfilter-policy>
    <name>test_policy</name>
    <action>activate</action>
</ipfilter-policy>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Deleting an IP Filter Policy

The following example uses the DELETE request to remove an IP filter policy from the device. You cannot delete a default IP filter policy. Note that there is no transactional state for an IP filter policy request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

DELETE *<base_URI>*/running/brocade-security/ipfilter-policy/name/<*policy_name*>

URI Request

```
DELETE https://10.10.10.10/rest/running/brocade-security/ipfilter-policy/name/46RB7ATw0XsERUO2X5h
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing all IP Filter Rules

The following example uses the GET request to display all IP filter rules defined on the device. You can view a specific rule in an IP filter policy configuration by specifying the rule in the request (for example, GET *<base_URI>*/running/brocade-security/ipfilter-rule/policy-name/fwd_ipv4/index/1).

Structure

GET *<base_URI>*/running/brocade-security/ipfilter-rule

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/ipfilter-rule
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<Response>
    <ipfilter-rule>
        <policy-name>default_ipv4</policy-name>
        <index>1</index>
```

```
            <source-ip>any</source-ip>
            <destination-start-port>22</destination-start-port>
            <destination-end-port>22</destination-end-port>
            <protocol>tcp</protocol>
            <permission>permit</permission>
            <destination-ip>any</destination-ip>
        </ipfilter-rule>
...
        <ipfilter-rule>
            <policy-name>46RB7ATw0XsERUO2X5h</policy-name>
            <index>1</index>
            <source-ip>177.171.183.217/30</source-ip>
            <destination-start-port>2</destination-start-port>
            <destination-end-port>2</destination-end-port>
            <protocol>udp</protocol>
            <permission>permit</permission>
            <destination-ip>243.112.59.242/0</destination-ip>
        </ipfilter-rule>
    </Response>
```

Configuring an IP Filter Rule

The following example uses the POST request to configure a new IP filter rule on the device. Note that there is no transactional state for an IP filter rule request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

POST *<base_URI>*/running/brocade-security/ipfilter-rule

URI Request

```
POST https://10.10.10.10/rest/running/brocade-security/ipfilter-rule
```

Request Body

```
<ipfilter-rule>
    <policy-name>fwd_ipv4</policy-name>
    <index>1</index>
    <source-ip>any</source-ip>
    <destination-start-port>1</destination-start-port>
    <destination-end-port>1024</destination-end-port>
    <protocol>tcp</protocol>
    <permission>permit</permission>
    <destination-ip>192.26.1.3</destination-ip>
</ipfilter-rule>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting an IP Filter Rule

The following example uses the DELETE request to remove an IP filter rule from the device. You cannot delete a default IP filter rule. Note that there is no transactional state for an IP filter rule request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

DELETE *<base_URI>*/running/brocade-security/ipfilter-rule/policy-name/*policy_name*/index/*<index_number>*

URI Request

```
DELETE https://10.10.10.10/rest/running/brocade-security/ipfilter-rule/policy-name/fwd_ipv4/index/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing Password Configurations for All Users on a Device

The following example uses the GET request to display password configurations for all users on the device.

Structure

GET *<base_URI>*/running/brocade-security/user-specific-password-cfg

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/user-specific-password-cfg
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <user-specific-password-cfg>
        <user-name>root</user-name>
        <hash-type>SHA512</hash-type>
    </user-specific-password-cfg>
    <user-specific-password-cfg>
        <user-name>admin</user-name>
        <hash-type>SHA512</hash-type>
    </user-specific-password-cfg>
    <user-specific-password-cfg>
        <user-name>user</user-name>
        <hash-type>SHA512</hash-type>
    </user-specific-password-cfg>
    <user-specific-password-cfg>
        <user-name>maintenance</user-name>
        <hash-type>SHA512</hash-type>
```

```
        </user-specific-password-cfg>
        <user-specific-password-cfg>
            <user-name>robertREST</user-name>
            <hash-type>SHA512</hash-type>
        </user-specific-password-cfg>
        <user-specific-password-cfg>
            <user-name>gumbo303</user-name>
            <hash-type>SHA512</hash-type>
        </user-specific-password-cfg>
    </Response>
```

Configuring a Password Configuration for a New User

The following example uses the POST request to configure a new password configuration for a user. You can use a PATCH request to edit an existing password configuration for a user.

Structure

POST *<base_URI>*/running/brocade-security/user-specific-password-cfg

URI Request

```
POST https://10.10.10.10/rest/running/brocade-security/user-specific-password-cfg
```

Request Body

```
<user-specific-password-cfg>
    <user-name>newUser1</user-name>
    <minimum-password-age>5</minimum-password-age>
    <maximum-password-age>30</maximum-password-age>
    <warn-on-expire>5</warn-on-expire>
    <enforce-expire>false</enforce-expire>
</user-specific-password-cfg>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting a Password Configuration for a User on a Device

The following example uses the DELETE request to remove a user's password configuration from the device.

Structure

DELETE *<base_URI>*/running/brocade-security/user-specific-password-cfg/user-name/<*user_name*>

URI Request

```
DELETE https://10.10.10.10/rest/running/brocade-security/user-specific-password-cfg/user-name/
newUser1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Deleting All Password User Policies

The following example uses the PATCH request to remove the password policies for all users.

Structure

PATCH *<base_URI>*/running/brocade-security/password-cfg

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-security/password-cfg
```

Request Body

No request body is required.

```
<password-cfg>
    <password-action>delete-all</password-action>
</password-cfg>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the Password Policy

The following example uses the GET request to display the password policy defined on a device.

Structure

GET *<base_URI>*/running/brocade-security/password-cfg

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/password-cfg
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <password-cfg>
        <hash-type>sha512</hash-type>
        <manual-hash-enabled>false</manual-hash-enabled>
        <minimum-length>8</minimum-length>
        <character-set>0</character-set>
        <user-name-allowed>true</user-name-allowed>
        <minimum-lower-case-character>0</minimum-lower-case-character>
        <minimum-upper-case-character>0</minimum-upper-case-character>
        <minimum-numeric-character>0</minimum-numeric-character>
        <minimum-special-character>0</minimum-special-character>
        <past-password-history>1</past-password-history>
        <minimum-password-age>0</minimum-password-age>
        <maximum-password-age>0</maximum-password-age>
        <warn-on-expire>0</warn-on-expire>
```

```
            <lock-out-threshold>0</lock-out-threshold>
            <lock-out-duration>30</lock-out-duration>
            <admin-lock-out-enabled>false</admin-lock-out-enabled>
            <repeat-character-limit>1</repeat-character-limit>
            <sequence-character-limit>1</sequence-character-limit>
            <password-config-changed>false</password-config-changed>
            <reverse-user-name-allowed>false</reverse-user-name-allowed>
            <minimum-difference>0</minimum-difference>
        </password-cfg>
    </Response>
```

Viewing the Password Policy for a Specific User

The following example uses the GET request to display the password policy for a specified user.

Structure

GET *<base_URI>*/running/brocade-security/user-specific-password-cfg/user-name/*user_name*

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/user-specific-password-cfg/user-name/Test1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <user-specific-password-cfg>
        <user-name>Robert</user-name>
        <hash-type>SHA512</hash-type>
        <minimum-password-age>20</minimum-password-age>
        <maximum-password-age>56</maximum-password-age>
        <warn-on-expire>14</warn-on-expire>
    </user-specific-password-cfg>
</Response>
```

Configuring the Password Policy

The following example uses the PATCH request to edit the current password configuration parameters. Note that while the example below edits all editable parameters, you can be more specific in the parameters you want to edit. For example, to edit only the repeat and sequence character limits, use the following request body:

```
<password-cfg>
    <repeat-character-limit>2</repeat-character-limit>
    <sequence-character-limit>2</sequence-character-limit>
</password-cfg>
```

Structure

PATCH *<base_URI>*/running/brocade-security/password-cfg

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-security/password-cfg
```

Request Body

```
<user-config>
    <name>Test1</name>
    <role>admin</role>
    <account-description/>
    <account-enabled>true</account-enabled>
    <password-change-enforced>false</password-change-enforced>
    <account-locked>false</account-locked>
    <home-virtual-fabric>1</home-virtual-fabric>
    <virtual-fabric-role-id-list>
        <role-id>admin=1-128</role-id>
    </virtual-fabric-role-id-list>
    <chassis-access-role/>
    <access-start-time/>
    <access-end-time/>
</user-config>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing User Configurations

The following example uses the GET request to display all user configurations on a device. You can view a specific user's configuration by specifying the user name in the request (for example, GET <base_URI>/running/brocade-security/user-config/testuser).

Structure

GET *<base_URI>*/running/brocade-security/user-config

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/user-config
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <user-config>
        <name>root</name>
        <role>root</role>
        <account-description>root</account-description>
        <account-enabled>true</account-enabled>
        <password-change-enforced>false</password-change-enforced>
        <account-locked>false</account-locked>
```

```
            <access-start-time/>
            <access-end-time/>
            <auth-token-present>false</auth-token-present>
        </user-config>
        <user-config>
            <name>admin</name>
            <role>admin</role>
            <account-description>Administrator</account-description>
            <account-enabled>true</account-enabled>
            <password-change-enforced>false</password-change-enforced>
            <account-locked>false</account-locked>
            <access-start-time/>
            <access-end-time/>
            <auth-token-present>false</auth-token-present>
        </user-config>
        <user-config>
            <name>user</name>
            <role>user</role>
            <account-description>User</account-description>
            <account-enabled>true</account-enabled>
            <password-change-enforced>false</password-change-enforced>
            <account-locked>false</account-locked>
            <access-start-time/>
            <access-end-time/>
            <auth-token-present>false</auth-token-present>
        </user-config>
        <user-config>
            <name>maintenance</name>
            <role>maintenance</role>
            <account-description>Maintenance</account-description>
            <account-enabled>false</account-enabled>
            <password-change-enforced>false</password-change-enforced>
            <account-locked>false</account-locked>
            <access-start-time/>
            <access-end-time/>
            <auth-token-present>false</auth-token-present>
        </user-config>

        <user-config>
            <name>gumbo303</name>
            <role>admin</role>
            <account-description/>
            <account-enabled>true</account-enabled>
            <password-change-enforced>false</password-change-enforced>
            <account-locked>false</account-locked>
            <access-start-time/>
            <access-end-time/>
            <auth-token-present>true</auth-token-present>
        </user-config>
    </Response>
```

Configuring a New User/Editing an Existing User

The following example uses the POST request to create a new user on the device. You can use the PATCH request to edit an existing user on the device.

Structure

POST *<base_URI>*/running/brocade-security/user-config

URI Request

```
POST https://10.10.10.10/rest/running/brocade-security/user-config
```

Request Body

```
<user-config>
       <name>new_user</name>
       <password>cGFzc3dvcmQ=</password>
       <account-description>User</account-description>
       <account-enabled>true</account-enabled>
       <home-virtual-fabric>128</home-virtual-fabric>
       <virtual-fabric-role-id-list>
       <role-id>user=128</role-id>
       </virtual-fabric-role-id-list>
       <chassis-access-role>user</chassis-access-role>
       <access-start-time/>
       <access-end-time/>
</user-config>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Viewing the Authentication Configuration

The following example uses the GET request to display the authentication mechanism for the device. For more information about the possible authentication mechanisms, refer to authentication-mode in the Parameters section of this module.

Structure

GET *<base_URI>*/running/brocade-security/auth-spec

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/auth-spec
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <auth-spec>
        <authentication-mode>local</authentication-mode>
```

```
            <primary-auth-log-messages>true</primary-auth-log-messages>
        </auth-spec>
    </Response>
```

Configuring the Authentication Configuration

The following example uses the PATCH request to make the following changes to authentication configuration:

- Change authentication-mode to 'radius;local'
- Set the activate-no-log-out to 'true' (no effect on existing sessions regardless of the chosen authentication mode)
- Set the primary-auth-log-messages to 'true' (log messages display for authentication failure)

Structure

PATCH *<base_URI>*/running/brocade-security/auth-spec

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-security/auth-spec
```

Request Body

No request body is required.

```
<auth-spec>
    <authentication-mode>radius;local</authentication-mode>
    <activate-no-log-out>true</activate-no-log-out>
    <primary-auth-log-messages>true</primary-auth-log-messages>
</auth-spec>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the RADIUS Server Configuration

The following example uses the GET request to display the current RADIUS server configuration.

Structure

GET *<base_URI>*/running/brocade-security/radius-server

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/radius-server
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <radius-server>
        <server>1.2.3.4</server>
        <port>1812</port>
        <timeout>3</timeout>
```

```
            <authentication>chap</authentication>
            <encryption-type>aes256</encryption-type>
            <position>1</position>
        </radius-server>
    </Response>
```

Configuring a RADIUS Server

The following example uses the POST request to configure a RADIUS server. You can use the PATCH request to edit an existing RADIUS server. This procedure assumes a knowledge of remote authentication as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Structure

POST *<base_URI>*/running/brocade-security/radius-server

URI Request

```
POST https://10.10.10.10/rest/running/brocade-security/radius-server
```

Request Body

```
<radius-server>
    <server>1.2.3.4</server>
    <port>1812</port>
    <timeout>3</timeout>
    <authentication>chap</authentication>
    <encryption-type>aes256</encryption-type>
</radius-server>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting a RADIUS Server

The following example uses the DELETE request to remove a RADIUS server from the device.

Structure

DELETE *<base_URI>*/running/brocade-security/radius-server/server/*server_name*

URI Request

```
DELETE https://10.10.10.10/rest/running/brocade-security/radius-server/server/1.2.3.4
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the TACACS+ Server Configuration

The following example uses the GET request to display the current TACACS+ server configuration.

Structure

GET *<base_URI>*/running/brocade-security/tacacs-server

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/tacacs-server
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <tacacs-server>
        <server>4.5.6.7</server>
        <port>49</port>
        <timeout>3</timeout>
        <authentication>chap</authentication>
        <encryption-type>aes256</encryption-type>
        <position>1</position>
    </tacacs-server>
    <tacacs-server>
        <server>1.2.3.4</server>
        <port>49</port>
        <timeout>3</timeout>
        <authentication>chap</authentication>
        <encryption-type>aes256</encryption-type>
        <position>2</position>
    </tacacs-server>
    <tacacs-server>
        <server>2.3.4.5</server>
        <port>49</port>
        <timeout>3</timeout>
        <authentication>chap</authentication>
        <encryption-type>aes256</encryption-type>
        <position>3</position>
    </tacacs-server>
    <tacacs-server>
        <server>6.7.8.9</server>
        <port>49</port>
        <timeout>3</timeout>
        <authentication>chap</authentication>
        <encryption-type>aes256</encryption-type>
        <position>4</position>
    </tacacs-server>
</Response>
```

Configuring a TACACS+ Server

The following example uses the POST request to configure a TACACS+ server. You can use the PATCH request to edit an existing TACACS+ server. This procedure assumes a knowledge of remote authentication as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Structure

POST *<base_URI>*/running/brocade-security/tacacs-server

URI Request

```
POST https://10.10.10.10/rest/running/brocade-security/tacacs-server
```

Request Body

```
<tacacs-server>
   <server>2.5.6.7</server>
   <port>49</port>
   <timeout>3</timeout>
   <authentication>chap</authentication>
   <encryption-type>aes256</encryption-type>
</tacacs-server>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting a TACACS+ Server

The following example uses the DELETE request to remove a TACACS server from the device.

Structure

DELETE *<base_URI>*/running/brocade-security/tacacs-server/server/*server_name*

URI Request

```
DELETE https://10.10.10.10/rest/running/brocade-security/tacacs-server/server/2.5.6.7
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the LDAP Server Configuration

The following example uses the GET request to display the current LDAP server configuration.

Structure

GET *<base_URI>*/running/brocade-security/ldap-server

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/ldap-server
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <ldap-server>
        <server>223.134.45.70</server>
        <port>38970</port>
        <domain>brocadcom.com</domain>
        <timeout>25</timeout>
        <tls-mode>starttls</tls-mode>
        <position>1</position>
    </ldap-server>
</Response>
```

Configuring a LDAP Server

The following example uses the POST request to configure a LDAP server. You can use the PATCH request to edit an existing LDAP server. This procedure assumes a knowledge of remote authentication as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Structure

POST *<base_URI>*/running/brocade-security/ldap-server

URI Request

```
POST https://10.10.10.10/rest/running/brocade-security/ldap-server
```

Request Body

```
<ldap-server>
    <server>2.2.3.4</server>
    <port>389</port>
    <domain>local</domain>
    <timeout>3</timeout>
    <tls-mode>starttls</tls-mode>
</ldap-server>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting a LDAP Server

The following example uses the DELETE request to remove a LDAP server from the device.

Structure

DELETE *<base_URI>*/running/brocade-security/ldap-server/*server_name*

URI Request

```
DELETE https://10.10.10.10/rest/running/brocade-security/ldap-server/2.2.3.4
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing LDAP Role Mappings

The following example uses the GET request to display the current LDAP role mappings. This procedure assumes a knowledge of remote authentication as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administrator's Guide*.

Structure

GET *<base_URI>*/running/brocade-security/ldap-role-map

URI Request

```
GET https://10.10.10.10/rest/running/brocade-security/ldap-role-map
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <ldap-role-map>
        <ldap-role>User1</ldap-role>
        <switch-role>admin=1-128</switch-role>
        <home-virtual-fabric>1</home-virtual-fabric>
        <chassis-access-role/>
    </ldap-role-map>
</Response>
```

Mapping LDAP Roles to Switch Roles

The following example uses the POST request to map the 'test_role4' LDAP role to 'admin' switch role. You can use a PATCH request to edit an existing LDAP mapping.

Structure

POST *<base_URI>*/running/brocade-security/ldap-role-map

URI Request

```
POST https://10.10.10.10/rest/running/brocade-security/ldap-role-map
```

Request Body

```
    <ldap-role-map>
        <ldap-role>test_role4</ldap-role>
        <switch-role>admin=1-110;user=111-115;operator=116;securityadmin=117-118</switch-role>
        <home-virtual-fabric>128</home-virtual-fabric>
        <chassis-access-role>admin</chassis-access-role>
    </ldap-role-map>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting a LDAP Role Mapping

The following example uses the DELETE request to remove the mapping between the 'test_role4' LDAP role from 'admin' switch role.

Structure

DELETE *<base_URI>*/running/brocade-security/ldap-role-map/ldap-role/<l*dap_role_name*>

URI Request

```
DELETE https://10.10.10.10/rest/running/brocade-security/ldap-role-map/ldap-role/test_role4
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the Security Crypto Configuration Templates

The following example uses the GET request to display SecCryptoCfg templates on the device. This procedure assumes a knowledge of SecCryptoCfg templates as used in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Structure

GET *<base_URI>*/running/brocade-security/sec-crypto-cfg-template

URI

```
GET https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg-template
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <sec-crypto-cfg-template>
        <name>default_cc</name>
        <template>[Ver]  0.2
[SSH]
Enc:aes128-cbc,aes256-cbc
Kex:diffie-hellman-group14-sha1,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521
Mac:hmac-sha1,hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
LDAP_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
RAD_Protocol:TLSv1.2
```

```
LDAP_Protocol:TLSv1.2
[LOG]
Syslog_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
Syslog_Protocol:TLSv1.2
[HTTPS]
Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
Protocol:TLSv1.2
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[X509v3]
Validation:Strict
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_strong</name>
        <template>[Ver]  0.2
[SSH]
Enc:aes128-ctr,aes192-ctr,aes256-ctr
Kex:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-
sha256,diffie-hellman-group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-
sha512,curve25519-sha256
Mac:hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4:!AEAD:!SHA1:!AESGCM
LDAP_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!
CAMELLIA:!CHACHA20:!SEED:!RC4:!AEAD:!SHA1:!AESGCM
RAD_Protocol:TLSv1.2
LDAP_Protocol:TLSv1.2
[LOG]
Syslog_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!
CAMELLIA:!CHACHA20:!SEED:!RC4:!AEAD:!SHA1:!AESGCM
Syslog_Protocol:TLSv1.2
[HTTPS]
Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4:!AEAD:!SHA1:!AESGCM
Protocol:TLSv1.3
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[X509v3]
Validation:Basic
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_fips</name>
        <template>[Ver]  0.2
[SSH]
Enc:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
Kex:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256
Mac:hmac-sha1,hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
LDAP_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
RAD_Protocol:TLSv1.2
LDAP_Protocol:TLSv1.2
```

```
[LOG]
Syslog_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
Syslog_Protocol:TLSv1.2
[HTTPS]
Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
Protocol:TLSv1.2
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[FIPS]
SelfTests:yes
BootProm:yes
Zeroize:yes
Inside:yes
[X509v3]
Validation:Basic
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_generic</name>
        <template>[Ver]  0.2
[SSH]
Enc:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
Kex:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-
sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1
Mac:hmac-sha1,hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4
LDAP_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!
CAMELLIA:!CHACHA20:!SEED:!RC4
RAD_Protocol:Any
LDAP_Protocol:Any
[LOG]
Syslog_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!
CAMELLIA:!CHACHA20:!SEED:!RC4
Syslog_Protocol:Any
[HTTPS]
Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4
Protocol:Any
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[X509v3]
Validation:Basic
</template>
    </sec-crypto-cfg-template>
</Response>
```

Viewing the Security Crypto Configuration

The following example uses the GET request to display the active security crypto configuration on the device.

Structure

GET *<base_URI>*/running/brocade-security/sec-crypto-cfg

URI

```
GET https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <sec-crypto-cfg>
        <ssh-cipher>aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc</
ssh-cipher>
        <ssh-kex>ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-
exchange-sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-
group1-sha1</ssh-kex>
        <ssh-mac>hmac-md5,hmac-sha1,hmac-sha2-256,hmac-sha2-512</ssh-mac>
        <https-cipher>!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM</https-cipher>
        <https-tlsv13-
cipher>TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256</https-tlsv13-
cipher>
        <radius-cipher>!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM</radius-cipher>
        <ldap-cipher>!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM</ldap-cipher>
        <syslog-cipher>!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM</syslog-cipher>
        <https-tls-protocol>any</https-tls-protocol>
        <radius-tls-protocol>any</radius-tls-protocol>
        <ldap-tls-protocol>any</ldap-tls-protocol>
        <syslog-tls-protocol>any</syslog-tls-protocol>
        <x509v3-validation-mode>basic</x509v3-validation-mode>
    </sec-crypto-cfg>
</Response>
```

Importing a SecCryptoCfg Template

The following example uses the PATCH request to import a template file from a specified external host. This procedure assumes a knowledge of SecCryptoCfg templates as used in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Note that the password used in this example is Base64 encoded.

Structure

PATCH *<base_URI>*/running/brocade-security/sec-crypto-cfg-template-action

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg-template-action
```

Request Body

```
<sec-crypto-cfg-template-action>
    <template-name>Test1</template-name>
    <action>import</action>
    <file-transfer-protocol-type>scp</file-transfer-protocol-type>
```

```
    <remote-user-name>rdpuser</remote-user-name>
    <remote-host-ip>10.10.10.10</remote-host-ip>
    <remote-user-password>U2lsYfraW4xMjM=</remote-user-password>
    <remote-directory>/directory1/directory2/BSI_config_plusGCM.txt</remote-directory>
</sec-crypto-cfg-template-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Exporting a SecCryptoCfg Template

The following example uses the PATCH request to exports a template file to the specified external host. This procedure assumes a knowledge of SecCryptoCfg templates as used in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Note that the password used in this example is Base64 encoded.

Structure

PATCH *<base_URI>*/running/brocade-security/sec-crypto-cfg-template-action

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg-template-action
```

Request Body

This request body shows an example using SCP.

```
<sec-crypto-cfg-template-action>
    <template-name>default_cc/template-name>
    <action>export</action>
    <remote-user-name>brocade</remote-user-name>
    <remote-host-ip>10.70.12.10<remote-host-ip>
    <remote-user-password>U2lsYfraW4xMjM=</remote-user-password>
    <remote-directory>/users/home40/brocade</remote-directory>
    <file-transfer-protocol-type>scp</file-transfer-protocol-type>
</sec-crypto-cfg-template-action>
```

This request body shows an example using FTP.

```
<sec-crypto-cfg-template-action>
     <template-name>default_cc</template-name>
  <action>export</action>
  <file-transfer-protocol-type>ftp</file-transfer-protocol-type>
  <remote-user-name>ftpuser</remote-user-name>
  <remote-host-ip>10.38.55.96</remote-host-ip>
  <remote-user-password>cdd2FuZXJh</remote-user-password>
  <remote-directory>dir1/dir2/my_template.txt</remote-directory>
 </sec-crypto-cfg-template-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Activating a SecCryptoCfg Template

You can activate a default or user-defined template files. The following example uses the PATCH request to activate the default 'default_cc' template file. This procedure assumes a knowledge of SecCryptoCfg templates as used in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Structure

PATCH *<base_URI>*/running/brocade-security/sec-crypto-cfg-template-action

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg-template-action
```

Request Body

```
<sec-crypto-cfg-template-action>
    <template-name>default_cc</template-name>
    <action>apply</action>
</sec-crypto-cfg-template-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Verifying a SecCryptoCfg Template

The following example uses the PATCH request to verify the running configuration against a required configuration specified in the template file. This procedure assumes a knowledge of SecCryptoCfg templates as used in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administrator's Guide*.

Structure

PATCH *<base_URI>*/running/brocade-security/sec-crypto-cfg-template-action

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg-template-action
```

Request Body

```
<sec-crypto-cfg-template-action>
    <template-name>default_strong</template-name>
    <action>verify</action>
</sec-crypto-cfg-template-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message. When the operation is not successful, the response has a “Verify failed” message body and a “400 Bad Request” status message.

Viewing the Allowed User and Rekey Interval Configuration

The following example uses the GET request to display the allowed user and rekey interval configuration.

Structure

GET *<base_URI>*/running/brocade-security/sshutil

URI

```
GET https://10.10.10.10/rest/running/brocade-security/sshutil
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <sshutil>
        <allow-user-name>admin</allow-user-name>
        <rekey-interval>90</rekey-interval>
    </sshutil>
</Response>
```

Configuring a User for Public Key and Rekey Interval Configuration

The following example uses a PATCH request to change the allowed user for sshutil operations and the rekey interval.

Structure

PATCH *<base_URI>*/running/brocade-security/sshutil

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sshutil
```

Request Body

```
<sshutil>
    <allow-user-name>admin</allow-user-name>
    <rekey-interval>0</rekey-interval>
</sshutil>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing Host Keys

The following example uses the GET request to display the host keys on the device.

Structure

GET *<base_URI>*/running/brocade-security/sshutil-key

URI

```
GET https://10.10.10.10/rest/running/brocade-security/sshutil-key
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <sshutil-key>
        <algorithm-type>rsa</algorithm-type>
        <key-type>host-key</key-type>
        <size>2048</size>
        <fingerprint>SHA256:bicvymiohvGI3Xec8YEwwURKoL8kO6LNIc7fArHwzsU</fingerprint>
    </sshutil-key>
    <sshutil-key>
        <algorithm-type>dsa</algorithm-type>
        <key-type>host-key</key-type>
        <size>1024</size>
        <fingerprint>SHA256:ShHm7+RWvHo5aSm0igGkbcBbW8JJIeB6Jrm9FH8SuXw</fingerprint>
    </sshutil-key>
    <sshutil-key>
        <algorithm-type>ecdsa</algorithm-type>
        <key-type>host-key</key-type>
        <size>256</size>
        <fingerprint>SHA256:o7hCS9vaMJVzw/lDkSlGGwVDSyYXdMt3c0A+GtNfH8c</fingerprint>
    </sshutil-key>
</Response>
```

Generating a Host Key

The following example uses the POST request to generate and install an 'ecdsa' Host key on the device.

Structure

POST *<base_URI>*/running/brocade-security/sshutil-key

URI

```
POST https://10.10.10.10/rest/running/brocade-security/sshutil-key
```

Request Body

```
<sshutil-key>
    <algorithm-type>ecdsa</algorithm-type>
    <key-type>host-key</key-type>
</sshutil-key>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status appears in the headers.

Deleting a Host Key

The following example uses the DELETE request to remove the 'rsa' Host key from the device.

Structure

DELETE *<base_URI>*/running/brocade-security/sshutil-key/

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/sshutil-key/
```

Request Body

```
{
  "sshutil-key": {
    "algorithm-type": "rsa",
    "key-type": "host-key"
  }
}
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Generating a Private Key

The following example uses the POST request to generate and install an 'dsa' private key on the device.

Note that the passphrase used in this example is Base64 encoded.

Structure

POST *<base_URI>*/running/brocade-security/sshutil-key

URI

```
POST https://10.10.10.10/rest/running/brocade-security/sshutil-key
```

Request Body

```
<sshutil-key>
    <algorithm-type>dsa</algorithm-type>
    <key-type>public-private-key</key-type>
</sshutil-key>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status appears in the headers.

Deleting a Private Key

The following example uses the DELETE request to remove a private key 'dsa' from the device.

Structure

DELETE *<base_URI>*/running/brocade-security/sshutil-key/

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/sshutil-key/
```

Request Body

```
{
  "sshutil-key": {
```

```
    "key-type": "public-private-key",
       "algorithm-type": "dsa"
  }
}
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing Public Keys

The following example uses the GET request to display public keys on a device.

Structure

GET *<base_URI>*/running/brocade-security/sshutil-public-key/user-name/<*user_name*>

URI

```
GET https://10.10.10.10/rest/running/brocade-security/sshutil-public-key/user-name/admin
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
{
    "Response": {
        "sshutil-public-key": {
            "user-name": "admin",
            "public-key": "ecdsa-sha2-nistp384
AAAAE2VjZHNhLXNoYTItbmlzdHAzODQAAAAIbmlzdHAzODQAAABhBHvXyLFlHfdrs4a6RDPZeGfGQqM/0GyEbXijWsrViLf0EOTQmEztrU/
OuxUSFMgwxXo4rXHCJ4pqvweyQU5e2n2PiKF8LWmJaFUNv+EkGByNyfYJDxP2lED6ni+zBPGD0w==
 root@int035025.englab.brocade
.com\nssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA9YcFcZTo/WU20BbMe6lRuXeU7hpeE8LakVxkZenDUKINsxEgZLKf/
Kf7Y6Q2NW6HG
M8bWjG0JfJJd9AEj4fMyANntArI6D7ShjAbH7dp0dDMGyN+B7D5TTI2COSFD6W9RBLU70Gq+68vZxjh4Uq
+r7aidhDbrLTWb4evvSi1h/Nz
7Q11DrPPqQdUnYALTK3zIC/W3W8PkwbVYq8bIaowtuBanFUd1k3VG/eBN6Ua7tGBQOe6OHNMowjxWu46IfXJ
+M6qIdsNIwontxhiOfn5fRh
p86U5JYlI12FyF7z7ovs3QJHO3D9VGQxqnL+qO+ltqtXk/b1HAGsOollhGaSqpw== root@int035025.englab.brocade.com
\n"
        }
    }
}
```

Deleting a Public Key for a User

The following example uses the DELETE request to remove all imported public keys for a single user 'testuser'.

Structure

DELETE *<base_URI>*/running/brocade-security/sshutil-public-key/user-name/<*user_name*>

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/sshutil-public-key/user-name/testuser
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Importing a Public Key from a Remote Host

The following example uses the PATCH request to import a public key from a remote host using the following parameters:

- action = import
- user-name = auserlocal
- remote-user-name = auserremote
- remote-host-ip = 11.11.11.11
- remote-user-password = U2lsYfraW4xMjM=
- remote-directory = ~auser/.ssh
- public-key-name = out_going.pub

Note that the password used in this example is Base64 encoded.

Structure

PATCH *<base_URI>*/running/brocade-security/sshutil-public-key-action

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sshutil-public-key-action
```

Request Body

```
<sshutil-public-key-action>
    <action>import</action>
    <user-name>auserlocal</user-name>
    <remote-user-name>auserremote</remote-user-name>
    <remote-host-ip>11.11.11.11</remote-host-ip>
    <remote-user-password>U2lsYfraW4xMjM=</remote-user-password>
    <remote-directory>~auser/.ssh</remote-directory>
    <public-key-name>in_coming.pub</public-key-name>
</sshutil-public-key-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Adding a Remote Host Key to a Known Host File

The following example uses the POST request to add a remote host key to a known host file.

Structure

POST *<base_URI>*/running/brocade-security/sshutil-known-host

URI

```
POST https://10.10.10.10/rest/running/brocade-security/sshutil-known-host
```

Request Body

```
<sshutil-known-host>
   <remote-host-name>10.155.54.69</remote-host-name>
   <fingerprint-list>
   <fingerprint>SHA256:cf+cnhsrbNfCSLPgqFCK5pet/TGXl9qCxNFWi0sAyik</fingerprint>
   </fingerprintlist>
</sshutil-known-host>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Exporting a Public Key to a Remote Host

Before you export a public key you must generate the private/public key pair on the local switch. The following example uses the PATCH request to export a public key to a remote host using the following parameters:

- action = export
- algorithm-type = rsa
- user-name = auserlocal
- remote-user-name = auserremote
- remote-host-ip = 11.11.11.11
- remote-user-password = U2lsYfraW4xMjM=
- remote-directory = ~auser/.ssh

Note that the password used in this example is Base64 encoded.

Structure

PATCH *<base_URI>*/running/brocade-security/sshutil-public-key-action

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sshutil-public-key-action
```

Request Body

```
<sshutil-public-key-action>
     <action>export</action>
     <algorithm-type>rsa</algorithm-type>
     <user-name>auserlocal</user-name>
     <remote-user-name>auserremote</remote-user-name>
     <remote-host-ip>11.11.11.11</remote-host-ip>
     <remote-user-password>U2lsYfraW4xMjM=</remote-user-password>
     <remote-directory>~auser/.ssh</remote-directory>
</sshutil-public-key-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring a User Password

The following example uses the PATCH request to change the password for a user.

Note that the password used in this example is Base64 encoded.

Structure

PATCH *<base_URI>*/running/brocade-security/password

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/password
```

Request Body

```
<password>
    <user-name>testuser</user-name>
    <old-password>cGFzc3dvcmQ=</old-password>
    <new-password>c3VubnlzaWRlMTIz</new-password>
</password>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Generating a Third-Party Security Certificate Request on the Switch

The following example uses the POST request to generate a third-party certificate request on the switch.

NOTE
If you are generating a web certificate (certificate-type = https) and there is an existing web certificate on the Fabric OS switch, an "400 Bad Request" response displays. You must delete the existing web certificate on the Fabric OS switch before generating a web certificate again.

Structure

POST *<base_URI>*/running/brocade-security/security-certificate-generate

URI

```
POST https://10.10.10.10/rest/running/brocade-security/security-certificate-generate
```

Request Body

```
<security-certificate-generate>
    <algorithm-type>rsa</algorithm-type>
    <certificate-entity>csr</certificate-entity>
    <certificate-type>https</certificate-type>
    <hash-type>sha256</hash-type>
    <key-size>2048</key-size>
    <years>10</years>
    <country-name>US</country-name>
    <state-name>Colorado</state-name>
    <locality-name>Broomfield</locality-name>
    <organization-name>Brocade</organization-name>
    <unit-name>SQA</unit-name>
    <domain-name>myswitch.brocade.com</domain-name>
    <ip-address-in-subject-alternative-name>0</ip-address-in-subject-alternative-name>
```

```
</security-certificate-generate>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Generating a Self-Signed Security Certificate on the Switch

The following example uses the POST request to generate a third-party certificate on the switch.

> **NOTE**
> If you are generating a web certificate (certificate-type = https) and there is an existing web certificate on the Fabric OS switch, an "400 Bad Request" response displays. You must delete the existing web certificate on the Fabric OS switch before generating a web certificate again.

Structure

POST *<base_URI>*/running/brocade-security/security-certificate-generate

URI

```
POST https://10.10.10.10/rest/running/brocade-security/security-certificate-generate
```

Request Body

```
<security-certificate-generate>
    <algorithm-type>rsa</algorithm-type>
    <certificate-entity>cert</certificate-entity>
    <certificate-type>https</certificate-type>
    <hash-type>sha256</hash-type>
    <key-size>2048</key-size>
</security-certificate-generate>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Importing a Security Certificate

The following example uses the PATCH request to import a certificate.

Note that the password used in this example is Base64 encoded.

Structure

PATCH *<base_URI>*/running/brocade-security/security-certificate-action/

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/security-certificate-action/
```

Request Body

This request body shows an example using SCP.

```
<security-certificate-action>
    <remote-user-name>TestUser</remote-user-name>
    <remote-host-ip>11.11.11.11</remote-host-ip>
    <remote-user-password>U2lsY67W4xMjM=</remote-user-password>
```

```
    <remote-directory>/directory1/directory2</remote-directory>
    <protocol>scp</protocol>
    <certificate-entity>ca-server</certificate-entity>
    <certificate-type>https</certificate-type>
    <certificate-name>Quad-Goodtertiary-Intermediate-root.pem</certificate-name>
    <operation>import</operation>
</security-certificate-action>
```

This request body shows an example using FTP.

```
<security-certificate-action>
    <remote-user-name>ftpuser</remote-user-name>
    <remote-host-ip>11.11.11.11</remote-host-ip>
    <remote-user-password>U2lsY67W4xMjM=</remote-user-password>
    <remote-directory>directory1/directory2/</remote-directory>
    <protocol>ftp</protocol>
    <certificate-entity>ca-server</certificate-entity>
    <certificate-type>https</certificate-type>
    <certificate-name>Quad-Goodtertiary-Intermediate-root.pem</certificate-name>
    <operation>import</operation>
</security-certificate-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Exporting a Security Certificate

The following example uses the PATCH request to export a certificate.

Note that the password used in this example is Base64 encoded.

Structure

PATCH *<base_URI>*/running/brocade-security/security-certificate-action/

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/security-certificate-action/
```

Request Body

This request body shows an example using SCP.

```
<security-certificate-action>
    <remote-user-name>Test2</remote-user-name>
    <remote-host-ip>11.11.11.11</remote-host-ip>
    <remote-user-password>U2lsYWRta56MjM=</remote-user-password>
    <remote-directory>/directory1/directory2/</remote-directory>
    <protocol>scp</protocol>
    <certificate-entity>csr</certificate-entity>
    <certificate-type>https</certificate-type>
    <operation>export</operation>
</security-certificate-action>
```

This request body shows an example using FTP.

```
<security-certificate-action>
    <remote-user-name>ftpuser</remote-user-name>
    <remote-host-ip>11.11.11.11</remote-host-ip>
    <remote-user-password>c2FuZ34XJh</remote-user-password>
    <remote-directory>directory1/directory2</remote-directory>
    <protocol>ftp</protocol>
    <certificate-entity>ca-client</certificate-entity>
    <certificate-type>fcap</certificate-type>
    <operation>export</operation>
</security-certificate-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Deleting a Security Certificate

The following example uses the DELETE request to delete security certificate from the device.

Structure

DELETE *<base_URI>*/running/brocade-security/security-certificate-action

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/security-certificate-action
```

Request Body

```
<security-certificate-action>
       <certificate-entity>ca-server</certificate-entity>
       <certificate-type>ldap</certificate-type>
</security-certificate-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing all Security Certificates on a Device

The following example uses the GET request to view all certificates on a switch '10.10.10.10'.

Structure

GET *<base_URI>*/running/brocade-security/security-certificate

URI

```
GET https://10.10.10.10/rest/running/brocade-security/security-certificate
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
```

```
<Response>
    <security-certificate>
        <certificate-entity>ca-client</certificate-entity>
        <certificate-type>commoncert</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-client</certificate-entity>
        <certificate-type>radius</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-client</certificate-entity>
        <certificate-type>ldap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-client</certificate-entity>
        <certificate-type>syslog</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-client</certificate-entity>
        <certificate-type>fcap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-server</certificate-entity>
        <certificate-type>https</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-server</certificate-entity>
        <certificate-type>radius</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-server</certificate-entity>
        <certificate-type>ldap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-server</certificate-entity>
        <certificate-type>syslog</certificate-type>
        <certificate/>
```

```
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>commoncert</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>https</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>radius</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>ldap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>syslog</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>fcap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>cert</certificate-entity>
        <certificate-type>commoncert</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>cert</certificate-entity>
        <certificate-type>https</certificate-type>
        <certificate>Issued To
    countryName               = US
    stateOrProvinceName       = California
    localityName              = San Jose
    organizationName          = Brocade
    organizationalUnitName    = Eng
```

```
    commonName                = 10.38.66.243
Issued By
    countryName               = US
    stateOrProvinceName       = California
    localityName              = San Jose
    organizationName          = Brocade
    organizationalUnitName    = Eng
    commonName                = 10.38.66.243
Period Of Validity
    Begins On                 Mar  2 18:28:08 2020 GMT
    Expires On                Mar  1 18:28:08 2025 GMT
Fingerprints
    SHA1 Fingerprint  98:13:36:DE:A7:8F:7E:1F:05:69:95:92:02:F4:3C:59:F9:3B:2D:D4
    SHA256 Fingerprint
 A6:9F:A4:0E:99:2A:1E:A5:2C:F0:F7:77:85:B2:61:2D:EC:B2:56:A0:43:48:5F:A2:05:3D:6D:04:22:36:ED:BB
Crypto Algorithm
    Signature Algorithm       sha256WithRSAEncryption
    Public Key Algorithm      rsaEncryption
    Public-Key                2048 bit
</certificate>
        <certificate-hexdump>-----BEGIN CERTIFICATE-----
MIIEejCCA2KgAwIBAgIUbXQr0pOicEo6R0PoGBUaGHIhDn8wDQYJKoZIhvcNAQEL
BQAwbDELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExETAPBgNVBAcT
CFNhbiBKb3NlMRAwDgYDVQQKEwdCcm9jYWRlMQwwCgYDVQQLEwNFbmcxFTATBgNV
BAMTDDEwLjM4LjY2LjI0MzAeFw0yMDAzMDIxODI4MDhaFw0yNTAzMDExODI4MDha
MGwxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMREwDwYDVQQHEwhT
YW4gSm9zZTEQMA4GA1UEChMHQnJvY2FkZTEMMAoGA1UECxMDRW5nMRUwEwYDVQQD
EwwxMC4zOC42Ni4yNDMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCt
WICnqujYuqx/0P2CIC7UPUlIcA5dSjwKRxhP+ZCzyj3cnwID0BH0OIY1uP/gbO+H
pYezmakXZ1jdPS7vRy/2tW2Y9Kt1sR2yVUEpYLH4p7xWXVImGjnQ3vTiGu5memn4
FacTioXze+SSQ2DdbcJICSuBt2pkNHs4Kyhl5kof5VbxGqT4MlzHf8ILsYmx2wDL
XiCp1td8amAESsGW6TgHn/R+SB77Q1gAtA85lMazbJSWwC4FIZk2P5getjPhOjXA
+unGFFonNLZTj0n94+qek2mjzFnMrAe9t7atu68Uef0q82bgIK8M+R6vZZ+g56tf
PGf4K44bXEQmn6C9Q1plAgMBAAGjggESMIIBDjAJBgNVHRMEAjAAMBEGCWCGSAGG
+EIBAQQEAwIGQDAdBgNVHQ4EFgQUkN/rw0+JLaiE5/rtuazaPUeCVVswgakGA1Ud
IwSBoTCBnoAUkN/rw0+JLaiE5/rtuazaPUeCVVuhcKRuMGwxCzAJBgNVBAYTAlVT
MRMwEQYDVQQIEwpDYWxpZm9ybmlhMREwDwYDVQQHEwhTYW4gSm9zZTEQMA4GA1UE
ChMHQnJvY2FkZTEMMAoGA1UECxMDRW5nMRUwEwYDVQQDEwwxMC4zOC42Ni4yNDOC
FG10K9KTonBKOkdD6BgVGhhyIQ5/MA4GA1UdDwEB/wQEAwICpDATBgNVHSUEDDAK
BggrBgEFBQcDATANBgkqhkiG9w0BAQsFAAOCAQEAR0FCAHIHbV6QTfZbJd6OilPw
NJnVqc9cSXL3G4q3wVHKIw3Yyd1Z9tIvMzIequZt4Fj8RnpbhgaxBnbPS80s6urv
6vl58lovIidmiEMfkQ39n/TFlLvwrKaZvZnA1Hx73pAMvC4TeZZFH9cEMBIrCEpU
FIbXlEAtDEeMJVU92WaBm6s0j+bu7RxFGqshHEk8Zga+DVdvcZb2XUXIE0kPIOgm
XcFShGKGiVbZWyS54HlbnP5buIsSTdb7wnJUOwWO2SVvyvnJ5VeiqCQw6lJIqAcO
h04YuY13lYhan7IWcb9W9GMr1wEcjpmd5S8zBU5HJNQXnxeV51xwgbP/KVY/2Q==
-----END CERTIFICATE-----
</certificate-hexdump>
    </security-certificate>
    <security-certificate>
        <certificate-entity>cert</certificate-entity>
        <certificate-type>radius</certificate-type>
        <certificate/>
        <certificate-hexdump/>
```

```
        </security-certificate>
        <security-certificate>
            <certificate-entity>cert</certificate-entity>
            <certificate-type>ldap</certificate-type>
            <certificate/>
            <certificate-hexdump/>
        </security-certificate>
        <security-certificate>
            <certificate-entity>cert</certificate-entity>
            <certificate-type>syslog</certificate-type>
            <certificate/>
            <certificate-hexdump/>
        </security-certificate>
        <security-certificate>
            <certificate-entity>cert</certificate-entity>
            <certificate-type>fcap</certificate-type>
            <certificate/>
            <certificate-hexdump/>
        </security-certificate>
    </Response>
```

Viewing the Extension Certificates

This example uses a GET request to view the Extension certificates on a switch.

Structure

GET <*base_URI*>/running/brocade-security/security-certificate-extension

URI

```
GET https://10.10.10.10/rest/running/brocade-security/security-certificate-extension
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <security-certificate-extension>
        <certificate-entity>csr</certificate-entity>
        <certificate-name>MyPKIcert.csr</certificate-name>
        <certificate>        Issued To:
            countryName               = US
            stateOrProvinceName       = Colorado
            localityName              = Broomfield
            organizationName          = Broadcom
            organizationalUnitName    = BSN
            commonName                = MyPKIcert
        Public Key Algorithm: id-ecPublicKey
        Public-Key: (384 bit)
        X509v3 Subject Key Identifier:
```

```
                A3:E5:5D:BC:43:D7:B6:D6:40:8E:F1:01:92:0E:80:97:TT:B4:4D:24
        X509v3 Key Usage: critical
                Digital Signature, Non Repudiation
</certificate>
        <certificate-hexdump>-----BEGIN CERTIFICATE REQUEST-----
MIIBojCCASkCAQAwajELMAkGA1UEBhMCVVMxETAPBgNVBAgMCENvbG9yYWRvMRMw
EQYDVQQHDApCcm9vbWZpZWxkMREwDwYDVQQKDAhCcm9hZGNvbTEMMAoGA1UECwwD
QlNOMRIwEAYDVQQDDAlNeVBLSWNlcnQwdjAQBgcqhkjOPQIBBgUrgQQAIgNiAARL
QDA+BgkqhkiG9w0BCQ4xMTAvMB0GA1UdDgQWBBTD5V28Q6e21kCN8QGSDoCX+rRN
JDAOBgNVHQ8BAf8EBAMCBsAwCgYIKoZIzj0EAwMDZwAwZAIwEa9an+99L3e2byXW
X8NlGTqRZrT6GPgihlU0T7jFxYUywDM++OyHcTdrm3DO5tQ2AjAIWvhVUIP8pMC8
ejQQomgmbY5H81sD8vxRV06eGYY0kiETBwITUCUP7F52ONQ+Lo8=
-----END CERTIFICATE REQUEST-----
</certificate-hexdump>
    </security-certificate-extension>
    <security-certificate-extension>
        <certificate-entity>cert</certificate-entity>
        <keypair-tag>MyPKIcert
</keypair-tag>
        <certificate-name>MyPKIcert.pem</certificate-name>
        <local-certificate>true</local-certificate>
        <certificate>Issued To
    countryName               = US
    stateOrProvinceName       = Colorado
    localityName              = Broomfield
    organizationName          = Broadcom
    organizationalUnitName    = BSN
    commonName                = MyPKIcert
Issued By
    countryName               = US
    stateOrProvinceName       = Colorado
    localityName              = Broomfield
    organizationName          = Broadcom
    organizationalUnitName    = BSN
    commonName                = MyPKIcert
Period Of Validity
    Begins On                 Apr  6 16:26:34 2020 GMT
    Expires On                Apr  5 16:26:34 2025 GMT
Fingerprints
    SHA1 Fingerprint  45:6Z:14:E2:X1:21:5F:80:68:E8:Y5:25:DC:CB:50:3F:77:78:BB:A0
    SHA256 Fingerprint
 A1:20:B8:54:4E:CD:28:84:26:52:4E:FG:39:4H:0F:20:D3:3F:8A:42:4A:89:F0:C2:29:C3:34:55:E9:D0:79:37
Crypto Algorithm
    Signature Algorithm       ecdsa-with-SHA384
    Public Key Algorithm      id-ecPublicKey
    Public-Key                384 bit
File-Hash        864fbc49f230a2538d11fab5c36345a16af4e395
</certificate>
        <certificate-hexdump>-----BEGIN CERTIFICATE-----
MIICVTCCAdugAwIBAgIURWT9rXL1irPg2iy1uqS9PK+Ir54wCgYIKoZIzj0EAwMw
ajELMAkGA1UEBhMCVVMxETAPBgNVBAgMCENvbG9yYWRvMRMwEQYDVQQHDApCcm9v
bWZpZWxkMREwDwYDVQQKDAhCcm9hZGNvbTEMMAoGA1UECwwDQlNOMRIwEAYDVQQD
DAlNeVBLSWNlcnQwHhcNMjAwNDA2MTYyNjM0WhcNMjUwNDA1MTYyNjM0WjBqMQsw
```

```
CQYDVQQGEwJVUzERMA8GA1UECAwIQ29sb3JhZG8xEzARBgNVBAcMCkJyb29tZmll
bGQxETAPBgNVBAoMCEJyb2FkY29tMQwwCgYDVQQLDANCU04xEjAQBgNVBAMMCU15
UEtJY2VydDB2MBAGByqGSM49AgEGBSuBBAAiA2IABEuFhidlCzGtkiiPoB4wbWMS
XbxDp7bWQI3xAZIOgJf6tE0kMB8GA1UdIwQYMBaAFMPlXbxDp7bWQI3xAZIOgJf6
tE0kMAoGCCqGSM49BAMDA2gAMGUCMD4CcrGSIrRHbflAfqF/R+qY23S/05nMymPl
B0LuC3pXwP1+XbRx+45ijV0hmtT8+AIxALZ6EbjNPpaco+V4j1JRnIV5gtDCzBj6
IVhGtyqGS1lbIz+9XHSRH30q/JFm12iVkw==
-----END CERTIFICATE-----
</certificate-hexdump>
    </security-certificate-extension>
</Response>
```

Viewing All IP Filter Rules

The following example uses the GET request to display all IP filter rules defined on the device. You can view a specific rule in an IP filter policy configuration by specifying the rule in the request (for example, GET *<base_URI>*/running/brocade-security/ipfilter-rule/policy-name/fwd_ipv4/index/1).

Structure

```
GET <base_URI>/running/brocade-security/ipfilter-rule
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/ipfilter-rule
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<Response>
    <ipfilter-rule>
        <policy-name>default_ipv4</policy-name>
        <index>1</index>
        <source-ip>any</source-ip>
        <destination-start-port>22</destination-start-port>
        <destination-end-port>22</destination-end-port>
        <protocol>tcp</protocol>
        <permission>permit</permission>
        <destination-ip>any</destination-ip>
    </ipfilter-rule>
...
    <ipfilter-rule>
        <policy-name>46RB7ATw0XsERUO2X5h</policy-name>
        <index>1</index>
        <source-ip>10.10.10.10/30</source-ip>
        <destination-start-port>2</destination-start-port>
        <destination-end-port>2</destination-end-port>
        <protocol>udp</protocol>
```

```
        <permission>permit</permission>
        <destination-ip>10.10.10.11/0</destination-ip>
    </ipfilter-rule>
</Response>
```

Configuring an IP Filter Policy

The following example uses the POST request to configure a new IP filter policy on the device. You can use a PATCH request to edit an existing IP policy filter. Note that there is no transactional state for an IP filter policy request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

```
POST <base_URI>/running/brocade-security/ipfilter-policy
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/ipfilter-policy
```

Request Body

```
<ipfilter-policy>
    <name>ipv4_new</name>
    <ip-version>IPv4</ip-version>
</ipfilter-policy>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Cloning an IP Filter Policy

The following example uses the PATCH request to clone IP filter policy test_policy with a new name destPolicy on the device. You can use a PATCH request to edit an existing IP policy filter. Note that there is no transactional state for an IP filter policy request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

```
 PATCH <base_URI>/running/brocade-security/ipfilter-policy
```

URI

```
PATCH  https://10.10.10.10/rest/running/brocade-security/ipfilter-policy
```

Request Body

```
<ipfilter-policy>
      <name>test_policy</name>
      <action>clone</action>
      <clone-destination-policy-name>destPolicy</clone-destination-policy-name>
</ipfilter-policy>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Activating an IP Filter Policy

The following example uses the PATCH request to activate the destPolicy IP filter policy on the device. Note that there is no transactional state for an IP filter policy request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

```
PATCH <base_URI>/running/brocade-security/ipfilter-policy
```

URI

```
PATCH  https://10.10.10.10/rest/running/brocade-security/ipfilter-policy
```

Request Body

```
<ipfilter-policy>
    <name>test_policy</name>
    <action>activate</action>
</ipfilter-policy>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Deleting an IP Filter Policy

The following example uses the DELETE request to remove an IP filter policy from the device. You cannot delete a default IP filter policy. Note that there is no transactional state for an IP filter policy request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

```
DELETE <base_URI>/running/brocade-security/ipfilter-policy/name/<policy_name>
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/ipfilter-policy/name/46RB7ATw0XsERUO2X5h
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing all IP Filter Policies

The following example uses the GET request to display all IP filter policies configured on the device. You can view a specific rule in a specific IP filter policy configuration by specifying the IP filter policy and index number in the request (for example, GET *<base_URI>*/running/brocade-security/ipfilter-policy/fwd_ipv4/index/1).

Structure

```
GET <base_URI>/running/brocade-security/ipfilter-policy
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/ipfilter-policy
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <ipfilter-policy>
        <name>default_ipv4</name>
        <ip-version>IPv4</ip-version>
        <is-policy-active>true</is-policy-active>
        <is-default-policy>true</is-default-policy>
    </ipfilter-policy>
    <ipfilter-policy>
        <name>default_ipv6</name>
        <ip-version>IPv6</ip-version>
        <is-policy-active>true</is-policy-active>
        <is-default-policy>true</is-default-policy>
    </ipfilter-policy>
    <ipfilter-policy>
        <name>fwd_ipv4</name>
        <ip-version>IPv4</ip-version>
        <is-policy-active>false</is-policy-active>
        <is-default-policy>false</is-default-policy>
    </ipfilter-policy>
    <ipfilter-policy>
        <name>46RB7ATw0XsERUO2X5h</name>
        <ip-version>IPv4</ip-version>
        <is-policy-active>false</is-policy-active>
        <is-default-policy>false</is-default-policy>
    </ipfilter-policy>
</Response>
```

Configuring an IP Filter Rule

The following example uses the POST request to configure a new IP filter rule on the device. Note that there is no transactional state for an IP filter rule request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

```
POST <base_URI>/running/brocade-security/ipfilter-rule
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/ipfilter-rule
```

Request Body

```
<ipfilter-rule>
    <policy-name>fwd_ipv4</policy-name>
    <index>1</index>
    <source-ip>any</source-ip>
    <destination-start-port>1</destination-start-port>
```

```
    <destination-end-port>1024</destination-end-port>
    <protocol>tcp</protocol>
    <permission>permit</permission>
    <destination-ip>192.26.1.3</destination-ip>
</ipfilter-rule>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting an IP Filter Rule

The following example uses the DELETE request to remove an IP filter rule from the device. You cannot delete a default IP filter rule. Note that there is no transactional state for an IP filter rule request in REST API like there is in CLI. Each request is saved before the REST API request is returned.

Structure

```
DELETE <base_URI>/running/brocade-security/ipfilter-rule/policy-name/policy_name/index/<index_number>
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/ipfilter-rule/policy-name/fwd_ipv4/index/1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing Password Configurations for All Users on a Device

The following example uses the GET request to display password configurations for all users on the device.

Structure

```
GET <base_URI>/running/brocade-security/user-specific-password-cfg
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/user-specific-password-cfg
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <user-specific-password-cfg>
        <user-name>admin</user-name>
        <hash-type>SHA512</hash-type>
    </user-specific-password-cfg>
    <user-specific-password-cfg>
```

```
        <user-name>user</user-name>
        <hash-type>SHA512</hash-type>
    </user-specific-password-cfg>
    <user-specific-password-cfg>
        <user-name>maintenance</user-name>
        <hash-type>SHA512</hash-type>
    </user-specific-password-cfg>
    <user-specific-password-cfg>
        <user-name>robertREST</user-name>
        <hash-type>SHA512</hash-type>
    </user-specific-password-cfg>
    <user-specific-password-cfg>
        <user-name>gumbo303</user-name>
        <hash-type>SHA512</hash-type>
    </user-specific-password-cfg>
</Response>
```

Configuring a Password Configuration for a New User

The following example uses the POST request to configure a new password configuration for a user. You can use a PATCH request to edit an existing password configuration for a user.

Structure

```
POST <base_URI>/running/brocade-security/user-specific-password-cfg
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/user-specific-password-cfg
```

Request Body

```
<user-specific-password-cfg>
    <user-name>newUser1</user-name>
    <minimum-password-age>5</minimum-password-age>
    <maximum-password-age>30</maximum-password-age>
    <warn-on-expire>5</warn-on-expire>
    <enforce-expire>false</enforce-expire>
</user-specific-password-cfg>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting a Password Configuration for a User on a Device

The following example uses the DELETE request to remove a user's password configuration from the device.

Structure

```
DELETE <base_URI>/running/brocade-security/user-specific-password-cfg/user-name/<user_name>
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/user-specific-password-cfg/user-name/newUser1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Deleting All Password User Policies

The following example uses the PATCH request to remove the password policies for all users.

Structure

```
PATCH <base_URI>/running/brocade-security/password-cfg
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/password-cfg
```

Request Body

```
<password-cfg>
    <password-action>delete-all</password-action>
</password-cfg>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Deleting All Password User Policies

The following example uses the PATCH request to remove the password policies for all users.

Structure

```
PATCH <base_URI>/running/brocade-security/password-cfg
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/password-cfg
```

Request Body

```
<password-cfg>
    <password-action>delete-all</password-action>
</password-cfg>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the Password Policy

The following example uses the GET request to display the password policy defined on a device.

Structure

```
GET <base_URI>/running/brocade-security/password-cfg
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/password-cfg
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <password-cfg>
        <hash-type>sha512</hash-type>
        <manual-hash-enabled>false</manual-hash-enabled>
        <minimum-length>8</minimum-length>
        <character-set>0</character-set>
        <user-name-allowed>true</user-name-allowed>
        <minimum-lower-case-character>0</minimum-lower-case-character>
        <minimum-upper-case-character>0</minimum-upper-case-character>
        <minimum-numeric-character>0</minimum-numeric-character>
        <minimum-special-character>0</minimum-special-character>
        <past-password-history>1</past-password-history>
        <minimum-password-age>0</minimum-password-age>
        <maximum-password-age>0</maximum-password-age>
        <warn-on-expire>0</warn-on-expire>
        <lock-out-threshold>0</lock-out-threshold>
        <lock-out-duration>30</lock-out-duration>
        <admin-lock-out-enabled>false</admin-lock-out-enabled>
        <repeat-character-limit>1</repeat-character-limit>
        <sequence-character-limit>1</sequence-character-limit>
        <password-config-changed>false</password-config-changed>
        <reverse-user-name-allowed>false</reverse-user-name-allowed>
        <minimum-difference>0</minimum-difference>
    </password-cfg>
</Response>
```

Viewing the Password Policy for a Specific User

The following example uses the GET request to display the password policy for a specified user.

Structure

```
GET <base_URI>/running/brocade-security/user-specific-password-cfg/user-name/user_name
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/user-specific-password-cfg/user-name/Test1
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <user-specific-password-cfg>
        <user-name>Robert</user-name>
        <hash-type>SHA512</hash-type>
        <minimum-password-age>20</minimum-password-age>
        <maximum-password-age>56</maximum-password-age>
        <warn-on-expire>14</warn-on-expire>
    </user-specific-password-cfg>
</Response>
```

Configuring the Password Policy

The following example uses the PATCH request to edit the current password configuration parameters. Note that while the example below edits all editable parameters, you can be more specific in the parameters you want to edit.

For example, to edit only the repeat and sequence character limits, use the following request body:

```
<password-cfg>
    <repeat-character-limit>2</repeat-character-limit>
    <sequence-character-limit>2</sequence-character-limit>
</password-cfg>
```

Structure

```
PATCH <base_URI>/running/brocade-security/password-cfg
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/password-cfg
```

Request Body

```
<user-config>
    <name>Test1</name>
    <role>admin</role>
    <account-description/>
    <account-enabled>true</account-enabled>
    <password-change-enforced>false</password-change-enforced>
    <account-locked>false</account-locked>
    <home-virtual-fabric>1</home-virtual-fabric>
    <virtual-fabric-role-id-list>
        <role-id>admin=1-128</role-id>
    </virtual-fabric-role-id-list>
    <chassis-access-role/>
    <access-start-time/>
    <access-end-time/>
</user-config>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing User Configurations

The following example uses the GET request to display all user configurations on a device. You can view a specific user's configuration by specifying the user name in the request (for example, `GET <base_URI>/running/brocade-security/user-config/testuser`).

Structure

```
GET <base_URI>/running/brocade-security/user-config
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/user-config
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <user-config>
        <name>admin</name>
        <role>admin</role>
        <account-description>Administrator</account-description>
        <account-enabled>true</account-enabled>
        <password-change-enforced>false</password-change-enforced>
        <account-locked>false</account-locked>
        <access-start-time/>
        <access-end-time/>
        <auth-token-present>false</auth-token-present>
    </user-config>
    <user-config>
        <name>user</name>
        <role>user</role>
        <account-description>User</account-description>
        <account-enabled>true</account-enabled>
        <password-change-enforced>false</password-change-enforced>
        <account-locked>false</account-locked>
        <access-start-time/>
        <access-end-time/>
        <auth-token-present>false</auth-token-present>
    </user-config>
    <user-config>
        <name>maintenance</name>
        <role>maintenance</role>
        <account-description>Maintenance</account-description>
        <account-enabled>false</account-enabled>
        <password-change-enforced>false</password-change-enforced>
        <account-locked>false</account-locked>
        <access-start-time/>
        <access-end-time/>
        <auth-token-present>false</auth-token-present>
```

```
    </user-config>
    <user-config>
        <name>gumbo303</name>
        <role>admin</role>
        <account-description/>
        <account-enabled>true</account-enabled>
        <password-change-enforced>false</password-change-enforced>
        <account-locked>false</account-locked>
        <access-start-time/>
        <access-end-time/>
        <auth-token-present>true</auth-token-present>
    </user-config>
</Response>
```

Configuring a New User/Editing an Existing User

The following example uses the POST request to create a new user on the device. You can use the PATCH request to edit an existing user on the device.

Structure

```
POST <base_URI>/running/brocade-security/user-config
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/user-config
```

Request Body

```
<user-config>
       <name>new_user</name>
       <password>cGFzc3dvcmQ=</password>
       <account-description>User</account-description>
       <account-enabled>true</account-enabled>
       <home-virtual-fabric>128</home-virtual-fabric>
       <virtual-fabric-role-id-list>
       <role-id>user=128</role-id>
       </virtual-fabric-role-id-list>
       <chassis-access-role>user</chassis-access-role>
       <access-start-time/>
       <access-end-time/>
</user-config>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Viewing the Authentication Configuration

The following example uses the GET request to display the authentication mechanism for the device. For more information about the possible authentication mechanisms, refer to authentication-mode in the Parameters section of this module.

Structure

```
GET <base_URI>/running/brocade-security/auth-spec
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/auth-spec
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <auth-spec>
        <authentication-mode>local</authentication-mode>
        <primary-auth-log-messages>true</primary-auth-log-messages>
    </auth-spec>
</Response>
```

Configuring the Authentication Configuration

The following example uses the PATCH request to make the following changes to authentication configuration:

- Change authentication-mode to 'radius;local'
- Set the activate-no-log-out to 'true' (no effect on existing sessions regardless of the chosen authentication mode)
- Set the primary-auth-log-messages to 'true' (log messages display for authentication failure)

Structure

```
PATCH <base_URI>/running/brocade-security/auth-spec
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/auth-spec
```

Request Body

```
<auth-spec>
    <authentication-mode>radius;local</authentication-mode>
    <activate-no-log-out>true</activate-no-log-out>
    <primary-auth-log-messages>true</primary-auth-log-messages>
</auth-spec>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the RADIUS Server Configuration

The following example uses the GET request to display the current RADIUS server configuration.

Structure

```
GET <base_URI>/running/brocade-security/radius-server
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/radius-server
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <radius-server>
        <server>1.2.3.4</server>
        <port>1812</port>
        <timeout>3</timeout>
        <authentication>chap</authentication>
        <encryption-type>aes256</encryption-type>
        <position>1</position>
    </radius-server>
</Response>
```

Configuring a RADIUS Server

The following example uses the POST request to configure a RADIUS server. You can use the PATCH request to edit an existing RADIUS server. This procedure assumes a knowledge of remote authentication as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Structure

```
POST <base_URI>/running/brocade-security/radius-server
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/radius-server
```

Request Body

```
<radius-server>
    <server>1.2.3.4</server>
    <port>1812</port>
    <timeout>3</timeout>
    <authentication>chap</authentication>
    <encryption-type>aes256</encryption-type>
</radius-server>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting a RADIUS Server

The following example uses the DELETE request to remove a RADIUS server from the device.

Structure

```
DELETE <base_URI>/running/brocade-security/radius-server/server/server_name
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/radius-server/server/1.2.3.4
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the TACACS+ Server Configuration

The following example uses the GET request to display the current TACACS+ server configuration.

Structure

```
GET <base_URI>/running/brocade-security/tacacs-server
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/tacacs-server
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <tacacs-server>
        <server>4.5.6.7</server>
        <port>49</port>
        <timeout>3</timeout>
        <authentication>chap</authentication>
        <encryption-type>aes256</encryption-type>
        <position>1</position>
    </tacacs-server>
    <tacacs-server>
        <server>1.2.3.4</server>
        <port>49</port>
        <timeout>3</timeout>
        <authentication>chap</authentication>
        <encryption-type>aes256</encryption-type>
        <position>2</position>
    </tacacs-server>
    <tacacs-server>
```

```
        <server>2.3.4.5</server>
        <port>49</port>
        <timeout>3</timeout>
        <authentication>chap</authentication>
        <encryption-type>aes256</encryption-type>
        <position>3</position>
    </tacacs-server>
    <tacacs-server>
        <server>6.7.8.9</server>
        <port>49</port>
        <timeout>3</timeout>
        <authentication>chap</authentication>
        <encryption-type>aes256</encryption-type>
        <position>4</position>
    </tacacs-server>
</Response>
```

Configuring a TACACS+ Server

The following example uses the POST request to configure a TACACS+ server. You can use the PATCH request to edit an existing TACACS+ server. This procedure assumes a knowledge of remote authentication as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Structure

```
POST <base_URI>/running/brocade-security/tacacs-server
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/tacacs-server
```

Request Body

```
<tacacs-server>
   <server>2.5.6.7</server>
   <port>49</port>
   <timeout>3</timeout>
   <authentication>chap</authentication>
   <encryption-type>aes256</encryption-type>
</tacacs-server>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting a TACACS+ Server

The following example uses the DELETE request to remove a TACACS server from the device.

Structure

```
DELETE <base_URI>/running/brocade-security/tacacs-server/server/server_name
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/tacacs-server/server/2.5.6.7
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the LDAP Server Configuration

The following example uses the GET request to display the current LDAP server configuration.

Structure

```
GET <base_URI>/running/brocade-security/ldap-server
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/ldap-server
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <ldap-server>
        <server>223.134.45.70</server>
        <port>38970</port>
        <domain>brocadcom.com</domain>
        <timeout>25</timeout>
        <tls-mode>starttls</tls-mode>
        <position>1</position>
    </ldap-server>
</Response>
```

Configuring a LDAP Server

The following example uses the POST request to configure a LDAP server. You can use the PATCH request to edit an existing LDAP server. This procedure assumes a knowledge of remote authentication as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Structure

```
POST <base_URI>/running/brocade-security/ldap-server
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/ldap-server
```

Request Body

```
<ldap-server>
    <server>2.2.3.4</server>
```

```
    <port>389</port>
    <domain>local</domain>
    <timeout>3</timeout>
    <tls-mode>starttls</tls-mode>
</ldap-server>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting a LDAP Server

The following example uses the DELETE request to remove a LDAP server from the device.

Structure

```
DELETE <base_URI>/running/brocade-security/ldap-server/server_name
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/ldap-server/2.2.3.4
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing LDAP Role Mappings

The following example uses the GET request to display the current LDAP role mappings. This procedure assumes a knowledge of remote authentication as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administrator's Guide*.

Structure

```
GET <base_URI>/running/brocade-security/ldap-role-map
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/ldap-role-map
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <ldap-role-map>
        <ldap-role>User1</ldap-role>
        <switch-role>admin=1-128</switch-role>
        <home-virtual-fabric>1</home-virtual-fabric>
        <chassis-access-role/>
```

```
    </ldap-role-map>
</Response>
```

Mapping LDAP Roles to Switch Roles

The following example uses the POST request to map the 'test_role4' LDAP role to 'admin' switch role. You can use a PATCH request to edit an existing LDAP mapping.

Structure

```
POST <base_URI>/running/brocade-security/ldap-role-map
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/ldap-role-map
```

Request Body

```
    <ldap-role-map>
        <ldap-role>test_role4</ldap-role>
        <switch-role>admin=1-110;user=111-115;operator=116;securityadmin=117-118</switch-role>
        <home-virtual-fabric>128</home-virtual-fabric>
        <chassis-access-role>admin</chassis-access-role>
    </ldap-role-map>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Deleting a LDAP Role Mapping

The following example uses the DELETE request to remove the mapping between the 'test_role4' LDAP role from 'admin' switch role.

Structure

```
DELETE <base_URI>/running/brocade-security/ldap-role-map/ldap-role/<ldap_role_name>
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/ldap-role-map/ldap-role/test_role4
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the Security Crypto Configuration Templates

The following example uses the GET request to display SecCryptoCfg templates on the device. This procedure assumes a knowledge of SecCryptoCfg templates as used in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Structure

```
GET <base_URI>/running/brocade-security/sec-crypto-cfg-template
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg-template
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <sec-crypto-cfg-template>
        <name>default_cc</name>
        <template>[Ver]  0.2
[SSH]
Enc:aes128-cbc,aes256-cbc
Kex:diffie-hellman-group14-sha1,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521
Mac:hmac-sha1,hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
LDAP_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
RAD_Protocol:TLSv1.2
LDAP_Protocol:TLSv1.2
[LOG]
Syslog_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
Syslog_Protocol:TLSv1.2
[HTTPS]
Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
Protocol:TLSv1.2
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[X509v3]
Validation:Strict
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_strong</name>
        <template>[Ver]  0.2
[SSH]
Enc:aes128-ctr,aes192-ctr,aes256-ctr
Kex:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-
hellman-group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,curve25519-sha256
Mac:hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4:!AEAD:!SHA1:!AESGCM
LDAP_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4:!AEAD:!SHA1:!AESGCM
RAD_Protocol:TLSv1.2
LDAP_Protocol:TLSv1.2
[LOG]
```

```
Syslog_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4:!AEAD:!SHA1:!AESGCM
Syslog_Protocol:TLSv1.2
[HTTPS]
Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4:!AEAD:!SHA1:!AESGCM
Protocol:TLSv1.3
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[X509v3]
Validation:Basic
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_fips</name>
        <template>[Ver]  0.2
[SSH]
Enc:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
Kex:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256
Mac:hmac-sha1,hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
LDAP_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
RAD_Protocol:TLSv1.2
LDAP_Protocol:TLSv1.2
[LOG]
Syslog_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
Syslog_Protocol:TLSv1.2
[HTTPS]
Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
Protocol:TLSv1.2
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[FIPS]
SelfTests:yes
BootProm:yes
Zeroize:yes
Inside:yes
[X509v3]
Validation:Basic
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_generic</name>
        <template>[Ver]  0.2
[SSH]
Enc:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
Kex:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-
hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1
Mac:hmac-sha1,hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4
LDAP_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4
```

```
RAD_Protocol:Any
LDAP_Protocol:Any
[LOG]
Syslog_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4
Syslog_Protocol:Any
[HTTPS]
Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4
Protocol:Any
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[X509v3]
Validation:Basic
</template>
    </sec-crypto-cfg-template>
</Response>
```

Viewing the Security Crypto Configuration

The following example uses the GET request to display the active security crypto configuration on the device.

Structure

GET *<base_URI>*/

```
running/brocade-security/sec-crypto-cfg
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <sec-crypto-cfg>
        <ssh-cipher>aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc</ssh-cipher>
        <ssh-kex>ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-
sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1</ssh-kex>
        <ssh-mac>hmac-md5,hmac-sha1,hmac-sha2-256,hmac-sha2-512</ssh-mac>
        <https-cipher>!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM</https-cipher>
        <https-tlsv13-cipher>TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256</
https-tlsv13-cipher>
        <radius-cipher>!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM</radius-cipher>
        <ldap-cipher>!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM</ldap-cipher>
        <syslog-cipher>!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM</syslog-cipher>
        <https-tls-protocol>any</https-tls-protocol>
        <radius-tls-protocol>any</radius-tls-protocol>
        <ldap-tls-protocol>any</ldap-tls-protocol>
        <syslog-tls-protocol>any</syslog-tls-protocol>
        <x509v3-validation-mode>basic</x509v3-validation-mode>
```

```
    </sec-crypto-cfg>
</Response>
```

Importing a SecCryptoCfg Template

The following example uses the PATCH request to import a template file from a specified external host. This procedure assumes a knowledge of SecCryptoCfg templates as used in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Note that the password used in this example is Base64 encoded.

Structure

```
PATCH <base_URI>/running/brocade-security/sec-crypto-cfg-template-action
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg-template-action
```

Request Body

```
<sec-crypto-cfg-template-action>
    <template-name>Test1</template-name>
    <action>import</action>
    <file-transfer-protocol-type>scp</file-transfer-protocol-type>
    <remote-user-name>rdpuser</remote-user-name>
    <remote-host-ip>10.10.10.10</remote-host-ip>
    <remote-user-password>U2lsYfraW4xMjM=</remote-user-password>
    <remote-directory>/directory1/directory2/BSI_config_plusGCM.txt</remote-directory>
</sec-crypto-cfg-template-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Exporting a SecCryptoCfg Template

The following example uses the PATCH request to exports a template file to the specified external host. This procedure assumes a knowledge of SecCryptoCfg templates as used in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Note that the password used in this example is Base64 encoded.

Structure

```
PATCH <base_URI>/running/brocade-security/sec-crypto-cfg-template-action
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg-template-action
```

Request Body

This request body shows an example using SCP.

```
<sec-crypto-cfg-template-action>
    <template-name>default_cc/template-name>
    <action>export</action>
```

```
    <remote-user-name>brocade</remote-user-name>
    <remote-host-ip>10.70.12.10<remote-host-ip>
    <remote-user-password>U2lsYfraW4xMjM=</remote-user-password>
    <remote-directory>/users/home40/brocade</remote-directory>
    <file-transfer-protocol-type>scp</file-transfer-protocol-type>
</sec-crypto-cfg-template-action>
```

This request body shows an example using FTP.

```
<sec-crypto-cfg-template-action>
     <template-name>default_cc</template-name>
  <action>export</action>
  <file-transfer-protocol-type>ftp</file-transfer-protocol-type>
  <remote-user-name>ftpuser</remote-user-name>
  <remote-host-ip>10.38.55.96</remote-host-ip>
  <remote-user-password>cdd2FuZXJh</remote-user-password>
  <remote-directory>dir1/dir2/my_template.txt</remote-directory>
 </sec-crypto-cfg-template-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Activating a SecCryptoCfg Template

You can activate a default or user-defined template files. The following example uses the PATCH request to activate the default 'default_cc' template file. This procedure assumes a knowledge of SecCryptoCfg templates as used in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administration Guide*.

Structure

```
PATCH <base_URI>/running/brocade-security/sec-crypto-cfg-template-action
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg-template-action
```

Request Body

```
<sec-crypto-cfg-template-action>
    <template-name>default_cc</template-name>
    <action>apply</action>
</sec-crypto-cfg-template-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Verifying a SecCryptoCfg Template

The following example uses the PATCH request to verify the running configuration against a required configuration specified in the template file. This procedure assumes a knowledge of SecCryptoCfg templates as used in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Administrator's Guide*.

Structure

```
PATCH <base_URI>/running/brocade-security/sec-crypto-cfg-template-action
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sec-crypto-cfg-template-action
```

Request Body

```
<sec-crypto-cfg-template-action>
    <template-name>default_strong</template-name>
    <action>verify</action>
</sec-crypto-cfg-template-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

When the operation is not successful, the response has a “Verify failed” message body and a “400 Bad Request” status message.

Viewing the Allowed User and Rekey Interval Configuration

The following example uses the GET request to display the allowed user and rekey interval configuration.

Structure

```
GET <base_URI>/running/brocade-security/sshutil
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/sshutil
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <sshutil>
        <allow-user-name>admin</allow-user-name>
        <rekey-interval>90</rekey-interval>
    </sshutil>
</Response>
```

Configuring a User for Public Key and Rekey Interval Configuration

The following example uses a PATCH request to change the allowed user for sshutil operations and the rekey interval.

Structure

```
PATCH <base_URI>/running/brocade-security/sshutil
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sshutil
```

Request Body

```
<sshutil>
    <allow-user-name>admin</allow-user-name>
    <rekey-interval>0</rekey-interval>
</sshutil>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Generating a Host Key

The following example uses the POST request to generate and install an 'ecdsa' Host key on the device.

Structure

```
POST <base_URI>/running/brocade-security/sshutil-key
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/sshutil-key
```

Request Body

```
<sshutil-key>
    <algorithm-type>ecdsa</algorithm-type>
    <key-type>host-key</key-type>
</sshutil-key>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status appears in the headers.

Deleting a Host Key

The following example uses the DELETE request to remove the 'rsa' Host key from the device.

Structure

```
DELETE <base_URI>/running/brocade-security/sshutil-key/
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/sshutil-key/
```

Request Body

```
{
  "sshutil-key": {
    "algorithm-type": "rsa",
    "key-type": "host-key"
  }
}
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Generating a Private Key

The following example uses the POST request to generate and install an 'dsa' private key on the device.

Note that the passphrase used in this example is Base64 encoded.

Structure

```
POST <base_URI>/running/brocade-security/sshutil-key
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/sshutil-key
```

Request Body

```
<sshutil-key>
    <algorithm-type>dsa</algorithm-type>
    <key-type>public-private-key</key-type>
</sshutil-key>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status appears in the headers.

Deleting a Private Key

The following example uses the DELETE request to remove a private key 'dsa' from the device.

Structure

```
DELETE <base_URI>/running/brocade-security/sshutil-key/
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/sshutil-key/
```

Request Body

```
{
  "sshutil-key": {
    "key-type": "public-private-key",
       "algorithm-type": "dsa"
  }
}
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing Public Keys

The following example uses the GET request to display public keys on a device.

Structure

```
GET <base_URI>/running/brocade-security/sshutil-public-key/user-name/<user_name>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/sshutil-public-key/user-name/admin
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
{
    "Response": {
        "sshutil-public-key": {
            "user-name": "admin",
            "public-key": "ecdsa-sha2-nistp384
AAAAE2VjZHNhLXNoYTItbmlzdHAzODQAAAAIbmlzdHAzODQAAABhBHvXyLFlHfdrs4a6RDPZeGfGQqM/0GyEbXijWsrViLf0EOTQmEztrU/
OuxUSFMgwxXo4rXHCJ4pqvweyQU5e2n2PiKF8LWmJaFUNv+EkGByNyfYJDxP2lED6ni+zBPGD0w== admin@int123456.englabs.brocade
.com\nssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA9YcFcZTo/WU20BbMe6lRuXeU7hpeE8LakVxkZenDUKINsxEgZLKf/Kf7Y6Q2NW6HGgu
M8bWjG0JfJJd9AEj4fMyANntArI6D7ShjAbH7dp07Q11DrPPqQdUnYALTK3zIC/W3W8PkwbVYq8bIaowtuBanFUd1k3VG/eBN6Ua7tGBQOe6O
p86U5JYlI12FyF7z7ovs3QJHO3D9VGQxqnL+qO+ltqtXk/b1HAGsOollhGaSqpw== admin@int123456.englabs.brocade.com\n"
        }
    }
}
```

Deleting a Public Key for a User

The following example uses the DELETE request to remove all imported public keys for a single user 'testuser'.

Structure

```
DELETE <base_URI>/running/brocade-security/sshutil-public-key/user-name/<user_name>
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/sshutil-public-key/user-name/testuser
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Importing a Public Key from a Remote Host

The following example uses the PATCH request to import a public key from a remote host using the following parameters:

- action = import
- user-name = auserlocal
- remote-user-name = auserremote
- remote-host-ip = 11.11.11.11
- remote-user-password = U2lsYfraW4xMjM=
- remote-directory = ~auser/.ssh
- public-key-name = out_going.pub

Note that the password used in this example is Base64 encoded.

Structure

```
PATCH <base_URI>/running/brocade-security/sshutil-public-key-action
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sshutil-public-key-action
```

Request Body

```
<sshutil-public-key-action>
    <action>import</action>
    <user-name>auserlocal</user-name>
    <remote-user-name>auserremote</remote-user-name>
    <remote-host-ip>11.11.11.11</remote-host-ip>
    <remote-user-password>U2lsYfraW4xMjM=</remote-user-password>
    <remote-directory>~auser/.ssh</remote-directory>
    <public-key-name>in_coming.pub</public-key-name>
</sshutil-public-key-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Adding a Remote Host Key to a Known Host File

The following example uses the POST request to add a remote host key to a known host file.

Structure

```
POST <base_URI>/running/brocade-security/sshutil-known-host
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/sshutil-known-host
```

Request Body

```
<sshutil-known-host>
   <remote-host-name>10.155.54.69</remote-host-name>
   <fingerprint-list>
   <fingerprint>SHA256:cf+cnhsrbNfCSLPgqFCK5pet/TGXl9qCxNFWi0sAyik</fingerprint>
   </fingerprintlist>
</sshutil-known-host>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Exporting a Public Key to a Remote Host

Before you export a public key you must generate the private/public key pair on the local switch. The following example uses the PATCH request to export a public key to a remote host using the following parameters:

- action = export
- algorithm-type = rsa
- user-name = auserlocal
- remote-user-name = auserremote
- remote-host-ip = 11.11.11.11
- remote-user-password = U2lsYfraW4xMjM=
- remote-directory = ~auser/.ssh

Note that the password used in this example is Base64 encoded.

Structure

```
PATCH <base_URI>/running/brocade-security/sshutil-public-key-action
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/sshutil-public-key-action
```

Request Body

```
<sshutil-public-key-action>
     <action>export</action>
     <algorithm-type>rsa</algorithm-type>
     <user-name>auserlocal</user-name>
     <remote-user-name>auserremote</remote-user-name>
     <remote-host-ip>11.11.11.11</remote-host-ip>
     <remote-user-password>U2lsYfraW4xMjM=</remote-user-password>
     <remote-directory>~auser/.ssh</remote-directory>
</sshutil-public-key-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring a User Password

The following example uses the PATCH request to change the password for a user.

Note that the password used in this example is Base64 encoded.

Structure

```
PATCH <base_URI>/running/brocade-security/password
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/password
```

Request Body

```
<password>
    <user-name>testuser</user-name>
    <old-password>cGFzc3dvcmQ=</old-password>
    <new-password>c3VubnlzaWRlMTIz</new-password>
</password>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Generating a Third-Party Security Certificate Request on the Switch

The following example uses the POST request to generate a third-party certificate request on the switch.

> **NOTE**
> If you are generating a web certificate (certificate-type = https) and there is an existing web certificate on the Fabric OS switch, an "400 Bad Request" response displays. You must delete the existing web certificate on the Fabric OS switch before generating a web certificate again.

Structure

```
POST <base_URI>/running/brocade-security/security-certificate-generate
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/security-certificate-generate
```

Request Body

```
<security-certificate-generate>
    <algorithm-type>rsa</algorithm-type>
    <certificate-entity>csr</certificate-entity>
    <certificate-type>https</certificate-type>
    <hash-type>sha256</hash-type>
    <key-size>2048</key-size>
    <years>10</years>
    <country-name>US</country-name>
    <state-name>Colorado</state-name>
    <locality-name>Broomfield</locality-name>
    <organization-name>Brocade</organization-name>
    <unit-name>SQA</unit-name>
    <domain-name>myswitch.brocade.com</domain-name>
</security-certificate-generate>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Generating a Self-Signed Security Certificate on the Switch

The following example uses the POST request to generate a third-party certificate on the switch.

> **NOTE**
> If you are generating a web certificate (certificate-type = https) and there is an existing web certificate on the Fabric OS switch, an "400 Bad Request" response displays. You must delete the existing web certificate on the Fabric OS switch before generating a web certificate again.

Structure

```
POST <base_URI>/running/brocade-security/security-certificate-generate
```

URI

```
POST https://10.10.10.10/rest/running/brocade-security/security-certificate-generate
```

Request Body

```
<security-certificate-generate>
    <algorithm-type>rsa</algorithm-type>
    <certificate-entity>cert</certificate-entity>
    <certificate-type>https</certificate-type>
    <hash-type>sha256</hash-type>
    <key-size>2048</key-size>
</security-certificate-generate>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status message.

Importing a Security Certificate

The following example uses the PATCH request to import a certificate.

Note that the password used in this example is Base64 encoded.

Structure

```
PATCH <base_URI>/running/brocade-security/security-certificate-action/certificate_name
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/security-certificate-action/Quad-Goodtertiary-
Intermediate-root.pem
```

Request Body

This request body shows an example using SCP.

```
<security-certificate-action>
    <remote-user-name>TestUser</remote-user-name>
    <remote-host-ip>11.11.11.11</remote-host-ip>
    <remote-user-password>U2lsY67W4xMjM=</remote-user-password>
    <remote-directory>/directory1/directory2</remote-directory>
    <protocol>scp</protocol>
    <certificate-entity>ca-server</certificate-entity>
    <certificate-type>https</certificate-type>
    <certificate-name>Quad-Goodtertiary-Intermediate-root.pem</certificate-name>
    <operation>import</operation>
</security-certificate-action>
```

This request body shows an example using FTP.

```
<security-certificate-action>
    <remote-user-name>ftpuser</remote-user-name>
    <remote-host-ip>11.11.11.11</remote-host-ip>
    <remote-user-password>U2lsY67W4xMjM=</remote-user-password>
    <remote-directory>directory1/directory2/</remote-directory>
    <protocol>ftp</protocol>
    <certificate-entity>ca-server</certificate-entity>
    <certificate-type>https</certificate-type>
    <certificate-name>Quad-Goodtertiary-Intermediate-root.pem</certificate-name>
    <operation>import</operation>
```

```
</security-certificate-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Exporting a Security Certificate

The following example uses the PATCH request to export a certificate.

Note that the password used in this example is Base64 encoded.

Structure

```
PATCH <base_URI>/running/brocade-security/security-certificate-action/certificate_name
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/security-certificate-action/Quad-Goodtertiary-
Intermediate-root.pem
```

Request Body

This request body shows an example using SCP.

```
<security-certificate-action>
    <remote-user-name>Test2</remote-user-name>
    <remote-host-ip>11.11.11.11</remote-host-ip>
    <remote-user-password>U2lsYWRta56MjM=</remote-user-password>
    <remote-directory>/directory1/directory2/</remote-directory>
    <protocol>scp</protocol>
    <certificate-entity>csr</certificate-entity>
    <certificate-type>https</certificate-type>
    <operation>export</operation>
</security-certificate-action>
```

This request body shows an example using FTP.

```
<security-certificate-action>
    <remote-user-name>ftpuser</remote-user-name>
    <remote-host-ip>11.11.11.11</remote-host-ip>
    <remote-user-password>c2FuZ34XJh</remote-user-password>
    <remote-directory>directory1/directory2</remote-directory>
    <protocol>ftp</protocol>
    <certificate-entity>ca-client</certificate-entity>
    <certificate-type>fcap</certificate-type>
    <operation>export</operation>
</security-certificate-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Deleting a Security Certificate

The following example uses the DELETE request to delete security certificate from the device.

Structure

```
DELETE <base_URI>/running/brocade-security/security-certificate-action
```

URI

```
DELETE https://10.10.10.10/rest/running/brocade-security/security-certificate-action
```

Request Body

```
<security-certificate-action>
       <certificate-entity>ca-server</certificate-entity>
       <certificate-type>ldap</certificate-type>
</security-certificate-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing all Security Certificates on a Device

The following example uses the GET request to view all certificates on a switch '10.10.10.10'.

Structure

```
GET <base_URI>/running/brocade-security/security-certificate
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/security-certificate
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <security-certificate>
        <certificate-entity>ca-client</certificate-entity>
        <certificate-type>commoncert</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-client</certificate-entity>
        <certificate-type>radius</certificate-type>
        <certificate/>
```

```
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-client</certificate-entity>
        <certificate-type>ldap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-client</certificate-entity>
        <certificate-type>syslog</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-client</certificate-entity>
        <certificate-type>fcap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-server</certificate-entity>
        <certificate-type>commoncert</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-server</certificate-entity>
        <certificate-type>https</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-server</certificate-entity>
        <certificate-type>radius</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-server</certificate-entity>
        <certificate-type>ldap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>ca-server</certificate-entity>
        <certificate-type>syslog</certificate-type>
        <certificate>Issued To
    countryName               = US
    stateOrProvinceName       = Colorado
    localityName              = Broomfield
    organizationName          = Brocade
    organizationalUnitName    = RootCA-num2
```

```
    commonName                = RootCA-num2@kali.org
    emailAddress              = RootCA-num2@kali.org
Issued By
    countryName               = US
    stateOrProvinceName       = Colorado
    localityName              = Broomfield
    organizationName          = Brocade
    organizationalUnitName    = RootCA-num2
    commonName                = RootCA-num2@kali.org
    emailAddress              = RootCA-num2@kali.org
Period Of Validity
Begins On  Apr 12 13:38:38 2017 GMT
Expires On  Apr  7 13:38:38 2037 GMT
Fingerprints
SHA1 Fingerprint  9A:CF:50:7B:8E:EE:E7:02:86:BA:C6:72:97:8A:47:CB:82:CB:16:72
SHA256 Fingerprint
 73:75:30:D1:5F:FE:F3:34:04:F5:DE:0B:6C:61:C6:6F:00:E5:9A:EA:C8:14:CD:78:66:74:B5:0D:7A:51:09:E5
</certificate>
        <certificate-hexdump>-----BEGIN CERTIFICATE-----
MIIGSzCCBDOgAwIBAgIJAKhg567O3fStMA0GCSqGSIb3DQEBCwUAMIGhMQswCQYD
VQQGEwJVUzERMA8GA1UECAwIQ29sb3JhZG8xEzARBgNVBAcMCkJyb29tZmllbGQx
EDAOBgNVBAoMB0Jyb2NhZGUxFDASBgNVBAsMC1Jvb3RDQS1udW0yMR0wGwYDVQQD
DBRSb290Q0EtbnVtMkBrYWxpLm9yZzEjMCEGCSqGSIb3DQEJARYUUm9vdENBLW51
bTJAa2FsaS5vcmcwHhcNMTcwNDEyMTMzODM4WhcNMzcwNDA3MTMzODM4WjCBoTEL
MAkGA1UEBhMCVVMxETAPBgNVBAgMCENvbG9yYWRvMRMwEQYDVQQHDApCcm9vbWZp
ZWxkMRAwDgYDVQQKDAdCcm9jYWRlMRQwEgYDVQQLDAtSb290Q0EtbnVtMjEdMBsG
A1UEAwwUUm9vdENBLW51bTJAa2FsaS5vcmcxIzAhBgkqhkiG9w0BCQEWFFJvb3RD
QS1udW0yQGthbGkub3JnMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA
w9accbTvcZn71BOoQ5ramvnFeVYqH8E7vww/yaz0yRWu5OPUhSf3quY4szIT3uPT
P9i/fSCaKP4X+KUmbFMU7ingvtcLQB6PYncAgtmTzS6T6QOu2+QQ2Z3FNINiFX4A
1tpE9Wj17iEZo4RJyoRQtYm+4MFqMW5FuVzJ7Eyu+gEt2AotzSkk3pTBJDTzY9mY
xCnIxTcRa3+QeYgSvDOfd++mNyXTcbBtJ4t698VSpeXZNJt2TQgje8p3Ylz+/l/+
AVlSw9vWbht9NRP9MXG9mxVS3IIhVMQKYansMEiUNqIbte/tQfPgXvd7tX04SXGY
VwEjIhShOQMSZMZSWUVmnlN9kuf8VWmcN+WjC6h3Ylmkh2E2b5vjtCygcUDgbwfE
2ahdh2S/Z7mkVaLLrKcYN+YV5JP7Wy7ODK8CgJwbI0ehMAtG6ITy/W6j/xFk4PPh
QeQ+wNKrwdbTJbPpGi9fa+tquxTKWPR6YtYY9Ym8mGERLKIdwSABkBWE/bSpiG3z
Hw3DTAgltISGWlSP1+1y+6FJwih6SAkAREFr5Y8HoJVDjCBiMD/raLGQzTfvNjYl
z6TXbMzaoWt+sGXdrQO1OxUWe0emd/H2lUK7U4GnjMnhNrBfGRQbHprEpzHDF5jf
SBqnIjiZ9BLIqIUHDrtTvW7aeuVqvg3qnaB+nFDuy+UCAwEAAaOBgzCBgDAdBgNV
HQ4EFgQUi4qnn3W+h/CHysetF2O8kGHBWuMwHwYDVR0jBBgwFoAUi4qnn3W+h/CH
ysetF2O8kGHBWuMwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAYYwHQYD
VR0lBBYwFAYIKwYBBQUHAwIGCCsGAQUFBwMBMA0GCSqGSIb3DQEBCwUAA4ICAQAM
Jf7uPugXrEty35Olb6wD8jgaikXDka2eKo4q/7YymVv8F7mZiEhoORrXYMqh8913
wZVPJL+aO0r4KcpSb7R+htxq+3se0I+mQkVT3GLWG/VUEwE+lsPLyiWx8ZLRxdCZ
EyLf+a0YGZvozgc8DArG8OT9LCxCZya3v8QhEeuqEnmvbqS/BYTgA+ioWLWIIOmy
k3ljdtcbqQBazk/+odP0jmvL8foEGloyWo+Hrt/GPxgHBTzn8olHe7WY20poj64B
I+J/EKOZ1nKCJUAHozwgEhvISqkulPz971HpKVH7kwMiHm4tKb2BUoYB9fffGkPt
MegsvaUmPBvViCpRBnQiK30vVIwmjYGOQ8tnjKF9uYupTmJHrlxyVqfGtiXYuFP3
eHhCAtP2tQLdpgNj8xktxc+JTQOmUuEzXfFToGEzUsAUAufcoxp2BqnBrubMcnRl
nYe0DMlIVjnu1DyvKDvByR4oUESQgTLFtuHVlQc4fVrkUnQ9JMbYBPDDfkqLQ9Ja
94ZF5bBdPgwybuslAFYRVrAlBIUevjlmVUDRCUhI/vzL+VXLestSqamQZ4RagubR
2AR0B+kT3A2BRXtgQbKz/+8PI5j0mxCri2y6AOfFtngslE3rfzuW+xfhv8WK+wc5
YbpLvMJ0V5SCo6VeS3Kd51l2bmeVqihmdv0SoyGGWQ==
```

```
-----END CERTIFICATE-----
</certificate-hexdump>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>commoncert</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>https</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>radius</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>ldap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>syslog</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>csr</certificate-entity>
        <certificate-type>fcap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>cert</certificate-entity>
        <certificate-type>commoncert</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>cert</certificate-entity>
        <certificate-type>https</certificate-type>
        <certificate>Issued To
    countryName               = US
    stateOrProvinceName       = California
    localityName              = San Jose
    organizationName          = Brocade
```

```
    organizationalUnitName    = Eng
    commonName                = 10.38.66.240
Issued By
    countryName               = US
    stateOrProvinceName       = California
    localityName              = San Jose
    organizationName          = Brocade
    organizationalUnitName    = Eng
    commonName                = 10.38.66.240
Period Of Validity
Begins On  Jun 29 20:50:57 2018 GMT
Expires On  Jun 28 20:50:57 2023 GMT
Fingerprints
SHA1 Fingerprint  C9:33:7F:4A:03:D9:70:E7:80:B3:00:C1:D3:9F:0E:E7:EC:31:0E:E1
SHA256 Fingerprint
 67:27:88:25:F1:C8:89:43:17:63:20:F1:6F:DD:F1:1E:93:B5:EE:E6:6E:16:56:44:DC:C7:D1:76:CD:D9:C4:E4
</certificate>
        <certificate-hexdump>-----BEGIN CERTIFICATE-----
MIIEZDCCA0ygAwIBAgIJAI5PTsYDEkaaMA0GCSqGSIb3DQEBCwUAMGwxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMREwDwYDVQQHEwhTYW4gSm9zZTEQ
MA4GA1UEChMHQnJvY2FkZTEMMAoGA1UECxMDRW5nMRUwEwYDVQQDEwwxMC4zOC42
Ni4yNDAwHhcNMTgwNjI5MjA1MDU3WhcNMjMwNjI4MjA1MDU3WjBsMQswCQYDVQQG
EwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxEDAO
BgNVBAoTB0Jyb2NhZGUxDDAKBgNVBAsTA0VuZzEVMBMGA1UEAxMMMTAuMzguNjYu
MjQwMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsrAcl8XDT4ozN4+K
RE6WvkVFrjOgB25MMqo9QDFXso83OhSEk1VTp0RorR+KI2lU+eNTqx0agNtFpk2x
3xx3o8BMH2uhNYkS38UbInaw+fzKAVULDiRa6yjH57kSmDqBPAXquWJMVw8f4Lcx
Yvff3/12Y+LmR4u7ipB/jPWEaNP11FnSBRyhrj1H3+d2oEdZAn9i4KxmNbKBkmYJ
rz+jACE03JbCJhWPai6XD5UIa+vaFovQA6K15Vehe5mDKKTonvTC1E2HgC5j4mLC
z9Nvx1AOlWhMdBACR5EgnnOJ3wcs0m8navm6iVEpE7rYm9wG30XuCxw0M1A8NIXf
axRoIwIDAQABo4IBBzCCAQMwCQYDVR0TBAIwADARBglghkgBhvhCAQEEBAMCBkAw
HQYDVR0OBBYEFNs38z31Jw9JK2hZmftj5od0Vu8LMIGeBgNVHSMEgZYwgZOAFNs3
8z31Jw9JK2hZmftj5od0Vu8LoXCkbjBsMQswCQYDVQQGEwJVUzETMBEGA1UECBMK
Q2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxEDAOBgNVBAoTB0Jyb2NhZGUx
DDAKBgNVBAsTA0VuZzEVMBMGA1UEAxMMMTAuMzguNjYuMjQwggkAjk9OxgMSRpow
DgYDVR0PAQH/BAQDAgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMA0GCSqGSIb3DQEB
CwUAA4IBAQCo0XRRimn2dgfuZxJWWfGfWoLkVkiHaj2IS3PvqpSerLV+MCqMU1Tc
k81dztg/ZHt9JhYqOuWnFVY3KW8QZqGGcwT785X1TyBCSdyT0AdqUB0Py8iy9zo/
BLVLJng4PDgRaEZEK/o+UHYdmOhqdzpzAAfFKQ7ZFf5lSTW9nLMpg8nFTYGuMbTu
77VP/zvNAxlszEVqMlmrY78symKOIDQnx/T9wrv4uPAx1ObaR3sR70BSlwtEt+3Q
Vxw7Z87N9lvXR8zQPInYZdRaAAYmp0QeVQy0YtRgqan/0KxwloomrKKhm/HUanho
ESoy3r1YNv8CVk5Zfa/8Pdr4yKuEe2tw
-----END CERTIFICATE-----
</certificate-hexdump>
    </security-certificate>
    <security-certificate>
        <certificate-entity>cert</certificate-entity>
        <certificate-type>radius</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>cert</certificate-entity>
```

```
        <certificate-type>ldap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>cert</certificate-entity>
        <certificate-type>syslog</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
    <security-certificate>
        <certificate-entity>cert</certificate-entity>
        <certificate-type>fcap</certificate-type>
        <certificate/>
        <certificate-hexdump/>
    </security-certificate>
</Response>
```

Viewing a List of Security Policies

This example uses a GET request to display a list of the security policies (SCC, FCS, and DCC).

Structure

```
GET <base_URI>/running/brocade-security/acl-policy
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/acl-policy
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
  <acl-policy>
    <policy-name>SCC_POLICY</policy-name>
    <policy-activated>FALSE</policy-activated>
    <policy-name>FCS_POLICY</policy-name>
    <policy-activated>FALSE</policy-activated>
  </acl-policy>
</Response>
```

Viewing the Defined DCC Policies

This example uses a GET request to display the device connection control (DCC) policies.

Structure

```
GET <base_URI>/running/brocade-security/defined-dcc-policy-member-list
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/defined-dcc-policy-member-list
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <defined-dcc-policy-member-list>
        <policy-name>DCC_POLICY_n</policy-name>
        <wwn>10:00:c4:f5:7c:16:a8:80</wwn>
        <member-type>switch</member-type>
        <domain-id>1</domain-id>
        <switch-name>sw0</switch-name>
        <ports>
            <port>1</port>
            <port>12</port>
            </ports>
    </defined-dcc-policy-member-list>
    <defined-dcc-policy-member-list>
        <policy-name>DCC_POLICY_n</policy-name>
        <wwn>30:0b:01:f5:7c:16:99:d4</wwn>
        <member-type>device</member-type>
    </defined-dcc-policy-member-list>
</Response>
```

Viewing the Defined FCS Policy

This example uses a GET request to display the defined fabric configuration server (FCS) policy.

Structure

```
GET <base_URI>/running/brocade-security/defined-fcs-policy-member-list
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/defined-fcs-policy-member-list
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <defined-fcs-policy-member-list>
        <switch-wwn>10:00:c4:f5:7c:16:a8:80</switch-wwn>
        <domain-id>1</domain-id>
        <switch-name>sw0</switch-name>
        <is-fcs-primary>1</is-fcs-primary>
        <fcs-position>TRUE</fcs-position>
```

```
    </defined-fcs-policy-member-list>
    <defined-fcs-policy-member-list>
        <switch-wwn>10:00:c4:f5:7c:16:a8:77</switch-wwn>
        <domain-id>7</domain-id>
        <switch-name>sw1</switch-name>
        <is-fcs-primary>FALSE</is-fcs-primary>
        <fcs-position>2</fcs-position>
    </defined-fcs-policy-member-list>
</Response>
```

Viewing the Defined SCC Policy

This example uses a GET request to display the defined switch connection control (SCC) policy.

Structure

```
GET <base_URI>/running/brocade-security/defined-scc-policy-member-list
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/defined-scc-policy-member-list
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <defined-scc-policy-member-list>
        <switch-wwn>10:00:c4:f5:7c:16:a8:80</switch-wwn>
        <domain-id>1</domain-id>
        <switch-name>sw0</switch-name>
    </defined-scc-policy-member-list>
    <defined-scc-policy-member-list>
        <switch-wwn>10:00:c4:f5:7c:16:a8:77</switch-wwn>
        <domain-id>7</domain-id>
        <switch-name>sw1</switch-name>
    </defined-scc-policy-member-list>
</Response>
```

Viewing a List of the Active DCC Policy Members

This example uses a GET request to display a list of the members of the active DCC policy.

Structure

```
GET <base_URI>/running/brocade-security/active-dcc-policy-member-list
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/active-dcc-policy-member-list
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <active-dcc-policy-member-list>
        <policy-name>DCC_POLICY_n</policy-name>
        <member-type>switch</member-type>
        <wwn>10:00:c4:f5:7c:16:a8:80</wwn>
        <domain-id>1</domain-id>
        <switch-name>sw0</switch-name>
             <ports>
                 <port>1</port>
                 <port>2</port>
            </ports>
    </active-dcc-policy-member-list>
    <active-dcc-policy-member-list>
        <policy-name>DCC_POLICY_n</policy-name>
        <member-type>device</member-type>
        <wwn>10:00:c4:f5:7c:16:a8:80</wwn>
    </active-dcc-policy-member-list>
</Response>
```

Viewing a List of the Active FCS Policy Members

This example uses a GET request to display a list of the members of the active FCS policy.

Structure

```
GET <base_URI>/running/brocade-security/active-fcs-policy-member-list
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/active-fcs-policy-member-list
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
<active-fcs-policy-member-list>
     <switch-wwn>10:00:c4:f5:7c:16:a8:80</switch-wwn>
<domain-id>1</domain-id>
<switch-name>sw0</switch-name>
<is-fcs-primary>TRUE</is-fcs-primary>
<fcs-position>1</fcs-position>
</active-fcs-policy-member-list>
<active-fcs-policy-member-list>
     <switch-wwn>10:00:c4:f5:7c:16:a8:77</switch-wwn>
```

```
<domain-id>7</domain-id>
<switch-name>sw1</switch-name>
<is-fcs-primary>FALSE</is-fcs-primary>
<fcs-position>2</fcs-position>
</active-fcs-policy-member-list>
</Response>
```

Viewing a List of the Active SCC Policy Members

This example uses a GET request to display a list of the members of the active SCCpolicy.

Structure

```
GET <base_URI>/running/brocade-security/active-scc-policy-member-list
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/active-scc-policy-member-list
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <active-scc-policy-member-list>
        <switch-wwn>10:00:c4:f5:7c:16:a8:80</switch-wwn>
        <domain-id>1</domain-id>
        <switch-name>sw0</switch-name>
    </active-scc-policy-member-list>
    <active-scc-policy-member-list>
        <switch-wwn>10:00:c4:f5:7c:16:a8:77</switch-wwn>
        <domain-id>7</domain-id>
        <switch-name>sw1</switch-name>
    </active-scc-policy-member-list>
</Response>
```

Viewing the Policy Distribution Configuration

This example uses a GET request to display fabric data distribution configuration parameters.

Structure

```
GET <base_URI>/running/brocade-security/policy-distribution-config
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/policy-distribution-config
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <policy-distribution-config>
        <scc-distribute-permission>accept</scc-distribute-permission>
        <dcc-distribute-permission>accept</dcc-distribute-permission>
        <password-distribute-permission>accept</password-distribute-permission>
        <fcs-distribute-permission>reject</fcs-distribute-permission>
        <auth-distribute-permission>accept</auth-distribute-permission>
        <ip-filter-distribute-permission>reject</ip-filter-distribute-permission>
        <fabric-wide-policies>
            <policy>tolerant-scc</policy>
            <policy>strict-dcc</policy>
            <policy>tolerant-fcs</policy>
        </fabric-wide-policies>
    </policy-distribution-config>
</Response>
```

Editing the Policy Distribution Configuration

This example uses a PATCH request to configure fabric data distribution configuration parameters.

Structure

```
GET <base_URI>/running/brocade-security/policy-distribution-config
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/policy-distribution-config
```

Request Body

There is no request body.

```
    <policy-distribution-config>
        <fcs-distribute-permission>reject</fcs-distribute-permission>
    </policy-distribution-config>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Viewing the Security Database Size

This example uses a GET request to display the size of the security database.

Structure

```
GET <base_URI>/running/brocade-security/security-policy-size
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/security-policy-size
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <security-policy-size>
        <define-size>59</define-size>
        <active-size>59</active-size>
        <max-size>1048576</max-size>
    </security-policy-size>
</Response>
```

Viewing the Security Statistics

This example uses a GET request to display the security statistics.

Structure

```
GET <base_URI>/running/brocade-security/security-violation-statistics
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/security-violation-statistics
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
     <security-violation-statistics>
        <domain-id>3</domain-id>
        <telnet-policy>23</telnet-policy>
        <http-policy>3</http-policy>
        <scc-policy>2</scc-policy>
        <dcc-policy>1</dcc-policy>
        <login>32</login>
        <authentication-fail>0</authentication-fail>
        <ts-out-sync>0</ts-out-sync>
        <illegal-cmd>0</illegal-cmd>
        <security-violation-statistics>
        <security-violation-statistics>
        <domain-id>7</domain-id>
        <telnet-policy>13</telnet-policy>
        <http-policy>0</http-policy>
        <scc-policy>30</scc-policy>
        <dcc-policy>32</dcc-policy>
        <login>31</login>
        <authentication-fail>19</authentication-fail>
        <ts-out-sync>0</ts-out-sync>
```

```
        <illegal-cmd>3</illegal-cmd>
    </security-violation-statistics>
</Response>
```

Viewing the DH-CHAP Shared Secret Key

This example uses a GET request to display the DH-CHAP shared secret key information from the database.

Structure

```
GET <base_URI>/running/brocade-security/dh-chap-authentication-secret
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/dh-chap-authentication-secret
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <dh-chap-authentication-secret>
        <switch-wwn>10:00:50:eb:1a:d0:50:00</switch-wwn>
        <domain-id>3</domain-id>
        <switch-name>sw0</switch-name>
   </dh-chap-authentication-secret>
    <dh-chap-authentication-secret>
        <device-wwn>10:00:50:eb:1a:d0:50:00</device-wwn>
    </dh-chap-authentication-secret>
</Response>
```

Viewing the Authentication Configuration

This example uses a GET request to display the authentication configuration.

Structure

```
GET <base_URI>/running/brocade-security/authentication-configuration
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/authentication-configuration
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <authentication-configuration>
```

```
        <authentication-type>all</authentication-type>
        <authentication-hash-type>all</authentication-hash-type>
        <dh-group>dh-group-all</dh-group>
        <switch-authentication-policy>passive</switch-authentication-policy>
        <device-authentication-policy>passive</device-authentication-policy>
    </authentication-configuration>
</Response>
```

Editing the Authentication Configuration

This example uses a PATCH request to configure the authentication configuration.

Structure

```
PATCH <base_URI>/running/brocade-security/authentication-configuration
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/authentication-configuration
```

Request Body

There is no request body.

```
    <authentication-configuration>
        <authentication-type>fcap</authentication-type>
    </authentication-configuration>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Viewing a List of the RBAC Permissions for the MOF Classes and CLI Commands

This example uses a GET request to display a list of the RBAC permissions for meta-object format (MOF) classes and for CLI commands.

Structure

```
GET <base_URI>/running/brocade-security/rbac-class
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/rbac-class
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <rbac-class>
        <class-name>RADIUS</class-name>
        <command-name>aaaconfig</command-name>
```

```
        <command-option>add</command-option>
        <permission>OM</permission>
        <context>chassis</context>
        <class-name>RADIUS</class-name>
        <command-name>aaaconfig</command-name>
        <command-option>authspec</command-option>
        <permission>OM</permission>
        <context>chassis</context>
        <class-name>RADIUS</class-name>
        <command-name>aaaconfig</command-name>
        <command-option>change</command-option>
        <permission>OM</permission>
        <context>chassis</context>
        <class-name>RADIUS</class-name>
        <command-name>aaaconfig</command-name>
        <command-option>move</command-option>
        <permission>OM</permission>
        <context>chassis</context>
        <class-name>RADIUS</class-name>
        <command-name>aaaconfig</command-name>
        <command-option>remove</command-option>
        <permission>OM</permission>
        <context>chassis</context>
        <class-name>RADIUS</class-name>
        <command-name>aaaconfig</command-name>
        <command-option>show</command-option>
        <permission>O</permission>
        <context>chassis</context>
    </rbac-class>
</Response>
```

Viewing the RBAC Permissions

This example uses a GET request to display all the RBAC permissions with context type details for the URLs.

Structure

```
GET <base_URI>/running/brocade-security/management-rbac-map
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/management-rbac-map
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "management-rbac-map": [
            {
                "url": "/rest/auth-token",
```

```
                "rbac-class": "none",
                "context-type": "both"
            },
            {
                "url": "/rest/brocade-module-version",
                "rbac-class": "none",
                "context-type": "both"
            },
            {
                "url": "/rest/login",
                "rbac-class": "none",
                "context-type": "both"
            },
            {
                "url": "/rest/logout",
                "rbac-class": "none",
                "context-type": "both"
            },
            {
                "url": "/rest/modules-state",
                "rbac-class": "none",
                "context-type": "both"
            },
            {
                "url": "/rest/operations/configdownload",
                "rbac-class": "ConfigManagement",
                "context-type": "both"
            },
            {
                "url": "/rest/operations/configupload",
                "rbac-class": "ConfigManagement",
                "context-type": "both"
            },
            ...
            },
            {
                "url": "/rest/running/brocade-zone/effective-configuration",
                "rbac-class": "Zoning",
                "context-type": "vf"
            },
            {
                "url": "/rest/running/brocade-zone/fabric-lock",
                "rbac-class": "Zoning",
                "context-type": "vf"
            },
            {
                "url": "/rest/running/brocade-zone/zone",
                "rbac-class": "Zoning",
                "context-type": "vf"
            }
        ]
    }
}
```

Viewing a List of Extension-related Security Certificates

This example uses a GET request to display a list of extension-related security certificates on a switch.

Structure

```
GET <base_URI>/running/brocade-security/security-certificate-extension
```

URI

```
GET https://10.10.10.10/rest/running/brocade-security/security-certificate-extension
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "security-certificate-extension": [
            {
                "certificate-entity": "csr",
                "certificate-name": "RESTCERT1.csr",
                "certificate": "        Issued To:\n            countryName               = US\n
 stateOrProvinceName       = Calif\n            localityName              = SJ\n            organizationName
         = Broc\n            organizationalUnitName    = Dev\n            commonName                = nprabudo
\n        Public Key Algorithm: id-ecPublicKey\n        Public-Key: (384 bit)\n        X509v3 Subject Key
 Identifier: \n                F3:3B:A3:33:33:03:E3:3A:33:BE:FA:E3:33:D3:33:3C:11:F3:0E:D0\n        X509v3 Key
 Usage: critical\n                Digital Signature, Non Repudiation\n",
                "certificate-hexdump": "-----BEGIN CERTIFICATE REQUEST-----
\nMIIBkzCCARkCAQAwWjELMAkGA1UEBhMCVGSM49AgEGBSuBBAAiA2IABJYAALBo7hxx6bxjqaPmaW9O
\n7TnnGGMxAKt1/WobvDENWJQKuTEltirBSrE/LLa1Obzu9heESeeD+mlVatCLTwy7\nrSw
+N1aNiEykpAsneRUPQQDAwNoADBlAjEAvcKKdT7ogwFqoskRWmwAcTEjutE9tNYeOJ+h\n
+e0dSE2ky8uEHmuYmlrN65PW6O1mAjBFGPAvGHL6D4CNFyviZhhG5od56pzHCN1m\nm5Xq0/TBw0hRUh1a/MfGV/v1Xes/cTM=\n-----END
 CERTIFICATE REQUEST-----\n",
                "certificate-verbose": "Certificate Request:\n    Data:\n        Version: 1 (0x0)\n
        Subject: C = US, ST = Calif, L = SJ, O = Broc, OU = Dev, CN = nprabudo\n        Subject
 Public Key Info:\n            Public Key Algorithm: id-ecPublicKey\n                Public-Key:
 (384 bit)\n                pub:\n                    04:44:00:00:b0:44:ee:1c:41:e4:bc:44:a4:a4:e4:
\n                    69:6f:4e:ed:39:e7:22:63:33:00:ab:75:fd:6a:1b:\n
                bc:31:0d:58:94:0a:b9:31:25:b6:2a:c1:4a:b1:3f:\n
            2c:b6:b5:39:bc:ee:f6:17:84:49:e7:83:fa:69:55:\n
        6a:d0:8b:2f:0c:bb:ad:2c:3e:37:54:8d:88:4c:a4:\n
 a4:0b:27:22:15:18:63:a1:d8:8a:aa:b1:42:33:b0:\n                    ef:f2:d1:ca:fe:b5:44\n
                ASN1 OID: secp384r1\n                NIST CURVE: P-384\n        Attributes:
\n        Requested Extensions:\n            X509v3 Subject Key Identifier: \n
        F3:9B:A6:33:46:04:E2:5A:55:BE:FA:E3:17:D2:88:8C:11:F3:0E:D0\n            X509v3
 Key Usage: critical\n                Digital Signature, Non Repudiation\n    Signature
 Algorithm: ecdsa-with-SHA384\n            30:65:02:22:00:bd:c2:8a:75:3e:e8:55:01:6a:a2:c9:11:5a:
\n         6c:00:71:31:23:ba:d1:3d:b4:d6:1e:38:9f:a1:f9:ed:1d:48:\n
      4d:a4:cb:cb:84:1e:6b:22:9a:5a:cd:eb:66:d6:e8:ed:66:02:\n
 30:45:18:f0:2f:18:72:fa:0f:80:8d:17:2b:e2:66:18:46:e6:\n
 87:79:ea:9c:c7:08:dd:66:9b:95:ea:d3:f4:c1:c3:48:51:52:\n         1d:5a:fc:c7:c6:57:fb:f5:5d:eb:3f:71:33\n"
            },
```

```
            {
                "certificate-entity": "cert",
                "keypair-tag": "RESTCERT1",
                "certificate-name": "RESTCERT1.pem",
                "local-certificate": true,
                "certificate": "Issued To \n    countryName               = US\n    stateOrProvinceName
      = Calif\n    localityName              = SJ\n    organizationName          = Broc\n
  organizationalUnitName    = Dev\n    commonName                = nprabudo\nIssued By \n
 countryName               = US\n    stateOrProvinceName       = Calif\n    localityName
        = SJ\n    organizationName          = Broc\n    organizationalUnitName    = Dev\n
 commonName                = nprabudo\nPeriod Of Validity \n    Begins On                 Nov  9
 10:30:38 2021 GMT\n    Expires On                Nov  8 10:30:38 2026 GMT\nFingerprints\n    SHA1
 Fingerprint  C3:44:8D:3A:D6:0D:F5:BF:CB:84:3E:FD:44:D7:61:D2:E1:32:01:E8\n    SHA256 Fingerprint
 5F:9B:B8:AF:E8:7D:67:33:63:D4:2C:96:A4:10:D8:56:7D:AE:FB:08:66:D6:ED:26:59:DB:B4:13:5C:E9:F2:BC\nCrypto
 Algorithm\n    Signature Algorithm      ecdsa-with-SHA384\n    Public Key Algorithm     id-ecPublicKey\n
 Public-Key               384 bit\nFile-Hash        a8b1d00e8fba3d548ca69442f3699e0b61207a18\n",
                "certificate-hexdump": "-----BEGIN CERTIFICATE-----
\nMIICNDCCAbugAwIBAgIUFf6POptrSE6CWes0eXQ9zUZc4NQwCgYIKoZIzj0EAwMw
\nWjELMAkGA1UEBhMCVVMxDjAMBgNVBAgMBUNhbGlmMQswCQYDVQQHDAJTSjENMAsG
\nA1UEDAVDYWxpZjELMAkGA1UEBwwCU0oxDTALBgNVBAoMBEJyb2MxDDAKBgNVBAsM
\nA0R1djERMA8GA1UEAwwIbnByYWJ1ZG8wdjAQBgcqhkjOPQIBBgUrgQQAIgNiAASWtTm8\n7vYXhEnng/
ppVWrQi08Mu60sPjO0DAfBgNVHSMEGDAWgBTz\nm6aVRgTiWoW++uMX0oiMEfMO0DAKBggqhkjOPQQDAwNnADBkAjBSIoSy7bXVPCJb
\nXdgca1Tr23zpfEruL+d3W+WW43YJNSpJBH46WSSDUtbWEgAcm60CMHgwXYt2iJIi\nYKifHBmY9zAEABtNX7Z8hEz
+tgp5AAdaBdKuApm5ow5w6EKqOd9Iig==\n-----END CERTIFICATE-----\n",
                "certificate-verbose": "Certificate:\n    Data:\n        Version: 3 (0x2)\n        Serial
 Number:\n            15:fe:8f:3a:9b:6b:22:4e:82:59:eb:99:79:74:3d:cd:46:5c:e0:d4\n        Signature
 Algorithm: ecdsa-with-SHA384\n        Issuer: C = US, ST = Calif, L = SJ, O = Broc, OU = Dev, CN = nprabudo
\n        Validity\n            Not Before: Nov  9 10:30:38 2021 GMT\n            Not After : Nov  8
 10:30:38 2026 GMT\n        Subject: C = US, ST = Calif, L = SJ, O = Broc, OU = Dev, CN = nprabudo\n
     Subject Public Key Info:\n            Public Key Algorithm: id-ecPublicKey\n                Public-
Key: (384 bit)\n                pub:\n                    04:55:00:00:b0:68:ee:1c:71:e9:bc:63:a9:a3:e6:
\n                    69:6f:4e:ed:39:e7:18:55:31:00:ab:75:fd:6a:1b:\n
                bc:31:0d:33:94:0a:b9:31:25:b6:2a:c1:4a:b1:3f:\n
            2c:b6:b5:39:bc:ee:f6:17:84:49:e7:83:fa:69:55:\n
       6a:d0:8b:4f:0c:bb:ad:2c:3e:37:56:8d:88:4c:a4:\n
 a4:0b:27:79:15:18:55:a1:d8:8a:aa:b1:42:16:b0:\n                    ef:f2:d1:ca:fe:b5:49\n
                ASN1 OID: secp384r1\n                NIST CURVE: P-384\n
 X509v3 extensions:\n            X509v3 Subject Key Identifier: \n
 F3:9B:A6:95:46:04:E2:5A:85:BE:FA:E3:17:D2:88:8C:11:F3:0E:D0\n            X509v3 Authority Key
 Identifier: \n                keyid:F3:9B:A6:95:46:04:E2:5A:85:BE:FA:E3:17:D2:88:8C:11:F3:0E:D0\n
 Signature Algorithm: ecdsa-with-SHA384\n         30:64:02:30:52:22:84:b2:ed:b5:d5:3c:22:5b:5d:d8:1c:6b:
\n         54:eb:db:7c:e9:7c:4a:ee:2f:e7:77:5b:e5:96:e3:76:09:35:\n
      2a:49:04:7e:3a:59:24:83:52:d6:d6:12:00:1c:9b:ad:02:30:\n
 78:30:5d:8b:76:88:92:22:60:a8:9f:1c:19:98:f7:30:04:00:\n
 1b:4d:5f:b6:7c:84:4c:fe:b6:0a:79:00:07:5a:05:d2:ae:02:\n         99:b9:a3:0e:70:e8:42:aa:39:df:48:8a\n"
            },
            {
                "certificate-entity": "ca",
                "certificate-name": "cacert-ECDSA.pem",
                "certificate": "Issued To \n    countryName               = US\n    stateOrProvinceName
      = Colorado\n    localityName              = Broomfield\n    organizationName
      = Brocade\n    organizationalUnitName    = SQA\n    commonName                =
 kali-2dash0.englabs.brocade.com\n    emailAddress              = tester@brocade.com\nIssued By
```

```
 \n    countryName               = US\n    stateOrProvinceName       = Colorado\n    localityName
             = Broomfield\n    organizationName          = Brocade\n    organizationalUnitName
    = SQA\n    commonName                = kali-2dash0.englabs.brocade.com\n    emailAddress
             = tester@brocade.com\nPeriod Of Validity \n    Begins On                 Jul 24
 14:25:17 2018 GMT\n    Expires On               Apr 19 14:25:17 2022 GMT\nFingerprints\n    SHA1
 Fingerprint  C2:04:E9:F1:71:2C:EA:9C:4C:59:81:BC:D4:7E:05:C7:6F:35:4B:36\n    SHA256 Fingerprint
 8A:1F:9A:34:D5:99:15:CE:EF:CF:8E:DE:E5:36:55:46:C1:08:CC:7D:95:17:01:57:BB:37:57:05:7C:60:FB:EB\nCrypto
 Algorithm\n    Signature Algorithm      ecdsa-with-SHA384\n    Public Key Algorithm     id-ecPublicKey\n
 Public-Key               384 bit\nFile-Hash        82c3ca0f62a3495623c188d9714c15592f9a4749\n",
                "certificate-hexdump": "-----BEGIN CERTIFICATE-----\nMIICyDCCAk
+gAwIBAgIJAJ2w3Rceg3B9MAoGCCqGSM49BAMDMIGgMQswCQYDVQQG
\nEwJVUzERMA8GA1UECAwIQ29sb3JhZG8xEzARBgNVBAcMCkJyb29tZmllbGQxEDAOyZG90\nMC5lbmdsYWIuYnJvY2FkZS5jb20xITAfBgkqhkiG9w0BCQ
\nYWRlLmNvbTAeFw0xODA3MjQxNDI1MTdaFw0yMjA0MTkxNDI1MTdaMIGg9lEPLCr
\nMhl2E20lKJ5IInpoFe4Ik8pmMWfLkhbpagWfieclN3Pd/8xFGroFqESWUkFQnAor
\n2GZMh43nPoprNqCUvO49a/0ekigJ3bQ5RqaEjrKNC5jypaNTMFEwHQYDVR0OBBYEUQbpOJitMA8GA1UdEwEB/
wQFMAMBAf8wCgYIKoZIzj0EAwMDZwAwZAIwHy5WvBWp\ns9zVL67ZaDCp4tFf7u566WUNU2ucn/c5JVQ7H4B/0/ifOq45ynjQx8hfAjB184Ib
\n7jc3h+4betSxwWM1efkRWKjZV4cDwrlSrYPwiwiN5uS5VHDoV+QYrK6NmQQ=\n-----END CERTIFICATE-----\n",
                "certificate-verbose": "Certificate:\n    Data:\n        Version: 3 (0x2)\n        Serial
 Number:\n            9d:b0:dd:17:1e:83:70:7d\n        Signature Algorithm: ecdsa-with-SHA384\n
 Issuer: C = US, ST = Colorado, L = Broomfield, O = Brocade, OU = SQA, CN = kali-2das0.englabs.brocade.com,
 emailAddress = tester@brocade.com\n        Validity\n            Not Before: Jul 24 14:25:17 2018 GMT\n
          Not After : Apr 19 14:25:17 2022 GMT\n        Subject: C = US, ST = Colorado, L = Broomfield,
 O = Brocade, OU = SQA, CN = kali-2das0.englabs.brocade.com, emailAddress = tester@brocade.com\n
   Subject Public Key Info:\n            Public Key Algorithm: id-ecPublicKey\n                Public-
Key: (384 bit)\n                pub:\n                    04:51:73:3a:6c:aa:55:d6:4d:37:d9:44:3c:b0:ab:
\n                    32:19:76:13:6d:25:33:9e:48:22:7a:68:15:ee:08:\n
                93:ca:66:31:67:cb:55:16:e9:6a:05:9f:89:e7:25:\n
           37:73:dd:ff:cc:45:1a:ba:05:a8:44:96:52:41:50:\n
       9c:0a:2b:d8:66:4c:22:8d:e7:3e:8a:6b:36:a0:94:\n
 bc:ee:3d:6b:fd:1e:92:55:09:dd:b4:39:46:a6:84:\n                    8e:b2:8d:0b:98:f2:a5\n                ASN1
 OID: secp384r1\n                NIST CURVE: P-384\n        X509v3 extensions:\n            X509v3 Subject Key
 Identifier: \n                C8:20:A7:84:83:B6:57:02:3B:7F:7B:75:7D:B7:51:06:E9:38:98:AD\n            X509v3
 Authority Key Identifier: \n                keyid:C8:20:A7:84:83:B6:57:02:3B:7F:7B:75:7D:B7:51:06:E9:38:98:AD
\n            X509v3 Basic Constraints: critical\n                CA:TRUE\n    Signature
 Algorithm: ecdsa-with-SHA384\n         30:64:02:30:1f:2e:56:bc:15:a9:b3:dc:d5:2f:ae:d9:68:30:
\n         a9:e2:d1:5f:ee:ee:7a:e9:65:0d:53:6b:9c:9f:f7:39:25:54:\n
      3b:1f:80:7f:d3:f8:9f:3a:ae:39:ca:78:d0:c7:c8:5f:02:30:\n
 75:f3:82:1b:ee:37:37:87:ee:1b:7a:d4:b1:c1:63:35:79:f9:\n
 11:58:a8:d9:57:87:03:c2:b9:52:ad:83:f0:8b:08:8d:e6:e4:\n        b9:54:70:e8:57:e4:18:ac:ae:8d:99:04\n"
            },
            {
                "certificate-entity": "csr",
                "certificate-name": "johnTestF_66-237.csr",
                "certificate": "        Issued To:\n            countryName               = US\n
          stateOrProvinceName       = Colorado\n            localityName              = Broomfield
\n            organizationName          = Brocade\n            organizationalUnitName    = SQA
\n            commonName                = johnTestF_66-237\n        Public Key Algorithm: id-
ecPublicKey\n        Public-Key: (384 bit)\n        X509v3 Subject Key Identifier: \n
 3C:CD:35:AD:F8:17:EB:29:C3:9C:37:E3:78:82:A6:57:5E:5B:5B:F2\n        X509v3 Key Usage: critical\n
    Digital Signature, Non Repudiation\n",
                "certificate-hexdump": "-----BEGIN CERTIFICATE REQUEST-----
\nMIIBqzCCATECAQAwcjELMAkGA1UEBhMCVVMxETAPBgNVBAgMCENvbG9yYWRvMRMw
\nEQYDVQQHDApCcm9vbWZpZWxkMRAwDgYDVQQKDAdCcm9jYWRlMQwwCgYDVQQLDANT
```

```
\nUUExGzAZBgNVBAMMEnJvYmswCZwDhVWrjuChbJmmlBcsmVz2VJ/O
\n5LrKANklFaBAMD4GCSqGSIb3DQEJDjExMC8wHQYDVR0OBBYEFDzNNa34F+spw5w3\n43iCpldeW1vyMA4GA1UdDwEB/
wQEAwIGwDAKBggqhkjOPQQDAwNoADBlAjB/ljZ0\nVtCxdM3QKyd+DEv2SGhBzJ4v3mW/4t3O9gJRmnOu9DLp4LMHuyFpQ1p
+GO8CMQC3\nO4WJJsuAGe9YWCy3qy+rPuGayf08yZmOX587zkrha9rPgncDz6EIfcn/1ChCJ4Y=\n-----END CERTIFICATE REQUEST-----
\n",
                "certificate-verbose": "Certificate Request:\n    Data:\n        Version: 1 (0x0)\n
     Subject: C = US, ST = Colorado, L = Broomfield, O = Brocade, OU = SQA, CN = johnTestF_66-237\n
    Subject Public Key Info:\n            Public Key Algorithm: id-ecPublicKey\n                Public-
Key: (384 bit)\n                pub:\n                    04:58:1f:9b:b0:f1:50:4a:d5:d7:58:65:ce:f9:3d:
\n                    35:59:6e:fe:6f:65:d5:78:66:3a:ff:97:c0:70:ea:\n
                1c:44:d2:e3:0f:c0:44:1d:c3:d8:64:b9:6c:bd:3a:\n
            29:87:10:22:d4:c7:6e:e9:e9:e5:03:87:bb:f0:81:\n
       23:c5:43:3d:4f:51:af:eb:30:09:9c:03:85:55:ab:\n
 8e:e0:a1:6c:99:a6:94:17:2c:99:5c:f6:54:9f:ce:\n                    e4:ba:ca:00:d9:25:15\n
                ASN1 OID: secp384r1\n                NIST CURVE: P-384\n        Attributes:
\n        Requested Extensions:\n            X509v3 Subject Key Identifier: \n
        3C:CD:35:AD:F8:17:EB:55:C3:9C:37:E3:78:82:A6:57:5E:5B:5B:F2\n            X509v3
 Key Usage: critical\n                Digital Signature, Non Repudiation\n    Signature
 Algorithm: ecdsa-with-SHA384\n         30:65:02:30:7f:96:36:74:56:d0:b1:74:cd:d0:2b:27:7e:0c:
\n         4b:f6:48:68:41:cc:9e:2f:de:55:bf:e2:dd:ce:f6:02:51:9a:\n
      73:ae:f4:32:e9:e0:b3:33:bb:21:69:43:5a:7e:18:ef:02:31:\n
 00:b7:3b:85:89:26:cb:80:19:ef:58:58:2c:b7:ab:2f:ab:3e:\n
 e1:9a:c9:fd:3c:c9:99:8e:5f:9f:3b:ce:4a:e1:6b:da:cf:82:\n         77:03:cf:a1:08:7d:c9:ff:d4:28:42:27:86\n"
            }
        ]
    }
}
```

brocade-snmp

This module provides a detailed view of the SNMP configurations used to monitor the switch through SNMP queries and trap notifications. FOS REST API version 8.2.1b and later support SNMPv1 and SNMPv3 queries and traps.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-snmp.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-snmp
  +--rw brocade-snmp
     +--rw system
     |  +--rw description?                 string
     |  +--rw location?                    string
     |  +--rw contact?                     string
     |  +--rw informs-enabled?             boolean
     |  +--rw encryption-enabled?          boolean
     |  +--rw audit-interval?              uint16
     |  +--rw default-config
     |  |  +--rw default-control*   snmp-types:default-control-type
     |  x--rw security-get-level?          snmp-types:security-level
     |  x--rw security-set-level?          snmp-types:security-level
     |  +--rw security-get-level-string?   snmp-types:security-level-string-type
     |  +--rw security-set-level-string?   snmp-types:security-level-string-type
     |  +--rw snmpv1-enabled?              boolean
     +--rw mib-capability* [mib-name]
     |  +--rw mib-name                 snmp-types:mibs-name
     |  +--rw is-mib-enabled-state?    boolean
     +--rw trap-capability* [trap-name]
     |  +--rw trap-name                 snmp-types:traps-name
     |  +--rw is-trap-enabled-state?    boolean
     |  +--rw severity?                 snmp-types:severity-level
     +--rw v1-account* [index]
     |  +--rw index              uint16
     |  +--rw community-name?    string
     |  +--rw community-group?   snmp-types:group-name
     +--rw v1-trap* [index]
     |  +--rw index                   uint16
     |  +--rw host?                   union
     |  +--rw trap-severity-level?    snmp-types:severity-level
     |  +--rw port-number?            inet:port-number
     +--rw v3-account* [index]
     |  +--rw index                       uint16
     |  +--rw user-name?                  string
     |  +--rw user-group?                 snmp-types:group-name
     |  +--rw authentication-protocol?    snmp-types:authentication-protocol-type
     |  +--rw privacy-protocol?           snmp-types:privacy-protocol-type
     |  +--rw authentication-password?    string
     |  +--rw privacy-password?           string
     |  +--rw manager-engine-id?          string
     +--rw v3-trap* [trap-index]
```

```
      |  +--rw trap-index             uint16
      |  +--rw usm-index?             uint16
      |  +--rw host?                  union
      |  +--rw trap-severity-level?   snmp-types:severity-level
      |  +--rw port-number?           inet:port-number
      |  +--rw informs-enabled?       boolean
      x--rw access-control* [index]
         x--rw index           uint16
         x--rw host?           union
         x--rw access-level?   snmp-types:access-permission
```

Examples

This section provides basic examples for using the brocade-operation-snmp module.

Viewing SNMP System Settings

The following example uses the GET request to view the SNMP system settings.

Structure

```
GET <base_URI>/running/brocade-snmp/system/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-snmp/system/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <system>
        <description>Fibre Channel Switch.</description>
        <location>End User Premise.</location>
        <contact>Field Support.</contact>
        <informs-enabled>false</informs-enabled>
        <security-get-level>0</security-get-level>
        <security-set-level>3</security-set-level>
        <security-get-level-string>no-security</security-get-level-string>
        <security-set-level-string>no-access</security-set-level-string>
        <audit-interval>60</audit-interval>
        <encryption-enabled>false</encryption-enabled>
        <snmpv1-enabled>true</snmpv1-enabled>
    </system>
</Response>
```

Configuring SNMP System Settings

The following example uses the PATCH request to set the following SNMP system settings:

- Informs = enabled
- Security SET level = 2
- audit interval = 90 minutes

Structure

```
PATCH <base_URI>/running/brocade-snmp/system/
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-snmp/system/
```

Request Body

```
<system>
    <informs-enabled>true</informs-enabled>
    <security-get-level-string>2</security-get-level-string>
    <audit-interval>90</audit-interval>
</system>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Viewing MIB Configurations

The following example uses the GET request to view the configuration for all MIBs.

Structure

```
GET <base_URI>/running/brocade-snmp/mib-capability/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-snmp/mib-capability/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <mib-capability>
        <mib-name>FE-MIB</mib-name>
        <is-mib-enabled-state>true</is-mib-enabled-state>
    </mib-capability>
    <mib-capability>
        <mib-name>SW-MIB</mib-name>
        <is-mib-enabled-state>true</is-mib-enabled-state>
    </mib-capability>
    <mib-capability>
        <mib-name>FA-MIB</mib-name>
```

```
        <is-mib-enabled-state>true</is-mib-enabled-state>
    </mib-capability>
    <mib-capability>
        <mib-name>FICON-MIB</mib-name>
        <is-mib-enabled-state>true</is-mib-enabled-state>
    </mib-capability>
    <mib-capability>
        <mib-name>HA-MIB</mib-name>
        <is-mib-enabled-state>true</is-mib-enabled-state>
    </mib-capability>
    <mib-capability>
        <mib-name>FCIP-MIB</mib-name>
        <is-mib-enabled-state>true</is-mib-enabled-state>
    </mib-capability>
    <mib-capability>
        <mib-name>IF-MIB</mib-name>
        <is-mib-enabled-state>true</is-mib-enabled-state>
    </mib-capability>
    <mib-capability>
        <mib-name>BROCADE-MAPS-MIB</mib-name>
        <is-mib-enabled-state>true</is-mib-enabled-state>
    </mib-capability>
    <mib-capability>
        <mib-name>T11-FC-ZONE-SERVER-MIB</mib-name>
        <is-mib-enabled-state>false</is-mib-enabled-state>
    </mib-capability>
</Response>
```

Enabling an MIB

The following example uses the PATCH request to disable the FCIP-MIB.

Structure

```
PATCH <base_URI>/running/brocade-snmp/mib-capability/<mib-name>/is-mib-enabled-state/<true|false>
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-snmp/mib-capability/fcip-mib/is-mib-enabled-state/true
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Disabling an MIB

The following example uses the PATCH request to disable the FE-MIB.

Structure

```
PATCH <base_URI>/running/brocade-snmp/mib-capability/<mib-name>/is-mib-enabled-state/<true|false>
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-snmp/mib-capability
```

Request Body

```
<mib-capability>
    <mib-name>FE-MIB</mib-name>
    <is-mib-enabled-state>false</is-mib-enabled-state>
</mib-capability>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the Configuration for Traps

The following example uses the GET request to view the configuration for all traps.

Structure

```
GET <base_URI>/running/brocade-snmp/trap-capability/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-snmp/trap-capability/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <trap-capability>
        <trap-name>swFCPortScn</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swEventTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    ...
    <trap-capability>
        <trap-name>t11ZsDefZoneChangeNotify</trap-name>
        <is-trap-enabled-state>false</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>t11ZsActivateNotify</trap-name>
        <is-trap-enabled-state>false</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
```

```
</Response>
```

Configuring a Trap

The following example uses the PATCH request to change the severity for swEventTrap to "warning".

Structure

```
PATCH <base_URI>/running/brocade-snmp/trap-capability/swEventTrap/severity/warning
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-snmp/trap-capability
```

Request Body

No request body is required.

```
<trap-capability>
    <trap-name>swEventTrap</trap-name>
    <severity>warning</severity>
</trap-capability>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the SNMPv1 Account Settings

The following example uses the GET request to view the SNMPv1 account settings.

Structure

```
GET <base_URI>/running/brocade-snmp/v1-account/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-snmp/v1-account/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <v1-account>
        <index>1</index>
        <community-group>read-write</community-group>
        <community-name>Secret C0de</community-name>
    </v1-account>
    <v1-account>
        <index>2</index>
        <community-group>read-write</community-group>
        <community-name>OrigEquipMfr</community-name>
```

```
    </v1-account>
    <v1-account>
        <index>3</index>
        <community-group>read-write</community-group>
        <community-name>private</community-name>
    </v1-account>
    <v1-account>
        <index>4</index>
        <community-group>read-only</community-group>
        <community-name>public</community-name>
    </v1-account>
    <v1-account>
        <index>5</index>
        <community-group>read-only</community-group>
        <community-name>common</community-name>
    </v1-account>
    <v1-account>
        <index>6</index>
        <community-group>read-only</community-group>
        <community-name>FibreChannel</community-name>
    </v1-account>
</Response>
```

Configuring a SNMPv1 Account

The following example uses the PATCH request to change the community name to 'common' for the SNMPv1 account '3'.

Structure

```
PATCH <base_URI>/running/brocade-snmp/v1-account/
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-snmp/v1-account/
```

Request Body

```
<v1-account>
   <index>3</index>
   <community-name>common</community-name>
</v1-account>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing SNMPv1 Trap Settings

The following example uses the GET request to view the SNMPv1 trap settings.

Structure

```
GET <base_URI>/running/brocade-snmp/v1-trap/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-snmp/v1-trap/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <v1-trap>
        <index>1</index>
        <host>0.0.0.0</host>
        <trap-severity-level>none</trap-severity-level>
        <port-number>162</port-number>
    </v1-trap>
    <v1-trap>
        <index>2</index>
        <host>0.0.0.0</host>
        <trap-severity-level>none</trap-severity-level>
        <port-number>162</port-number>
    </v1-trap>
    <v1-trap>
        <index>3</index>
        <host>0.0.0.0</host>
        <trap-severity-level>none</trap-severity-level>
        <port-number>162</port-number>
    </v1-trap>
    <v1-trap>
        <index>4</index>
        <host>0.0.0.0</host>
        <trap-severity-level>none</trap-severity-level>
        <port-number>162</port-number>
    </v1-trap>
    <v1-trap>
        <index>5</index>
        <host>0.0.0.0</host>
        <trap-severity-level>none</trap-severity-level>
        <port-number>162</port-number>
    </v1-trap>
    <v1-trap>
        <index>6</index>
        <host>0.0.0.0</host>
        <trap-severity-level>none</trap-severity-level>
        <port-number>162</port-number>
    </v1-trap>
</Response>
```

Configuring a SNMPv1 Trap

The following example uses the PATCH request to change the severity level for index 1 to 'warning' and the host IP address for index 5 to '10.10.10.10'.

Structure

```
PATCH <base_URI>/running/brocade-snmp/v1-trap/
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-snmp/v1-trap/
```

Request Body

```
<v1-trap>
    <index>2</index>
    <trap-severity-level>warning</trap-severity-level>
</v1-trap>
<v1-trap>
    <index>5</index>
    <host>10.10.10.10</host>
</v1-trap>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the SNMPv3 Account Settings

The following example uses the GET request to view the SNMPv3 account settings.

Structure

```
GET <base_URI>/running/brocade-snmp/v3-account/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-snmp/v3-account/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <v3-account>
        <index>1</index>
        <user-name>snmpadmin1</user-name>
        <user-group>read-write</user-group>
        <authentication-protocol>noauth</authentication-protocol>
        <privacy-protocol>nopriv</privacy-protocol>
        <authentication-password/>
        <privacy-password/>
```

```
        <manager-engine-id>00:00:00:00:00:00:00:00:00:00</manager-engine-id>
    </v3-account>
    <v3-account>
        <index>2</index>
        <user-name>snmpadmin2</user-name>
        <user-group>read-write</user-group>
        <authentication-protocol>noauth</authentication-protocol>
        <privacy-protocol>nopriv</privacy-protocol>
        <authentication-password/>
        <privacy-password/>
        <manager-engine-id>00:00:00:00:00:00:00:00:00:00</manager-engine-id>
    </v3-account>
    ...
    <v3-account>
        <index>6</index>
        <user-name>admin</user-name>
        <user-group>read-only</user-group>
        <authentication-protocol>noauth</authentication-protocol>
        <privacy-protocol>nopriv</privacy-protocol>
        <authentication-password/>
        <privacy-password/>
        <manager-engine-id>00:00:00:00:00:00:00:00:00:00</manager-engine-id>
    </v3-account>
</Response>
```

Configuring a SNMPv3 Account

The following example uses the PATCH request to change the manager engine for index 1 to '10:10:10:00:00:00:00:00'.

Structure

```
PATCH <base_URI>/running/brocade-snmp/v3-account/
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-snmp/v3-account/
```

Request Body

```
<v3-account>
    <index>1</index>
    <user-name>snmpadmin1</user-name>
    <manager-engine-id>10:10:10:00:00:00:00:00:00:00</manager-engine-id>
</v3-account>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing SNMPv3 Trap Settings

The following example uses the GET request to view the SNMPv3 trap settings.

Structure

```
GET <base_URI>/running/brocade-snmp/v3-trap/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-snmp/v3-trap/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <v3-trap>
        <trap-index>1</trap-index>
        <usm-index>6</usm-index>
        <host>10.10.10.11</host>
        <port-number>162</port-number>
        <trap-severity-level>informational</trap-severity-level>
        <informs-enabled>false</informs-enabled>
    </v3-trap>
    <v3-trap>
        <trap-index>2</trap-index>
        <usm-index>2</usm-index>
        <host>0.0.0.0</host>
        <port-number>162</port-number>
        <trap-severity-level>none</trap-severity-level>
        <informs-enabled>false</informs-enabled>
    </v3-trap>
    <v3-trap>
        <trap-index>3</trap-index>
        <usm-index>3</usm-index>
        <host>0.0.0.0</host>
        <port-number>162</port-number>
        <trap-severity-level>none</trap-severity-level>
        <informs-enabled>false</informs-enabled>
    </v3-trap>
    <v3-trap>
        <trap-index>4</trap-index>
        <usm-index>4</usm-index>
        <host>0.0.0.0</host>
        <port-number>162</port-number>
        <trap-severity-level>none</trap-severity-level>
        <informs-enabled>false</informs-enabled>
    </v3-trap>
    <v3-trap>
        <trap-index>5</trap-index>
        <usm-index>5</usm-index>
        <host>0.0.0.0</host>
        <port-number>162</port-number>
        <trap-severity-level>none</trap-severity-level>
```

```
        <informs-enabled>false</informs-enabled>
    </v3-trap>
    <v3-trap>
        <trap-index>6</trap-index>
        <usm-index>6</usm-index>
        <host>10.10.10.12</host>
        <port-number>162</port-number>
        <trap-severity-level>informational</trap-severity-level>
        <informs-enabled>false</informs-enabled>
    </v3-trap>
</Response>
```

Configuring a SNMPv3 Trap

The following example uses the PATCH request to configure the USM index to '5' and the host to '10.10.10.10' for trap index 6.

Structure

```
PATCH <base_URI>/running/brocade-snmp/v3-trap/
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-snmp/v3-trap/
```

Request Body

```
<v3-trap>
   <trap-index>6</trap-index>
   <usm-index>5</usm-index>
   <host>10.10.10.10</host>
</v3-trap>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing SNMP Access Control Settings

The following example uses the GET request to view the SNMP access control settings.

Structure

```
GET <base_URI>/running/brocade-snmp/access-control/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-snmp/access-control/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <access-control>
        <index>0</index>
        <host>0.0.0.0</host>
        <access-level>rw</access-level>
    </access-control>
    <access-control>
        <index>1</index>
        <host>0.0.0.0</host>
        <access-level>rw</access-level>
    </access-control>
    <access-control>
        <index>2</index>
        <host>0.0.0.0</host>
        <access-level>rw</access-level>
    </access-control>
    <access-control>
        <index>3</index>
        <host>0.0.0.0</host>
        <access-level>rw</access-level>
    </access-control>
    <access-control>
        <index>4</index>
        <host>0.0.0.0</host>
        <access-level>rw</access-level>
    </access-control>
    <access-control>
        <index>5</index>
        <host>0.0.0.0</host>
        <access-level>rw</access-level>
    </access-control>
</Response>
```

Configuring SNMP Access Control Settings

The following example uses the PATCH request to configure the host IP address to '0.0.0.11' for index 2.

Structure

```
PATCH <base_URI>/running/brocade-snmp/access-control/
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-snmp/access-control/
```

Request Body

```
<access-control>
   <index>2</index>
   <host>0.0.0.11</host>
</access-control>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

brocade-supportlink

This module enables you to display the existing Support Link configuration and history, enable Support Link, and configure Support Link.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-supportlink.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-supportlink
  +--rw brocade-supportlink
     +--rw supportlink-profile
     |  +--rw supportlink-enabled?       boolean
     |  +--rw server?                    inet:host
     |  +--rw port?                      inet:port-number
     |  +--rw user-name?                 union
     |  +--rw start-date?                union
     |  +--rw period?                    uint32
     |  +--rw start-time?                uint32
     |  +--rw end-time-period?           uint32
     |  +--ro random-start-time?         string
     |  +--rw retry-time?                uint32
     |  +--rw collection-time?           string
     |  +--rw proxy-user?                string
     |  +--rw proxy-password?            brocade-security-type:base64-string-type
     |  +--rw proxy-server?              union
     |  +--rw proxy-port?                inet:port-number
     |  +--rw proxy-protocol?            enumeration
     |  +--rw group-tag?                 string
     |  +--ro next-collection-time?      union
     |  +--ro next-service-start-time?   union
     |  +--ro next-service-retry-time?   yang:date-and-time
     |  +--ro last-collection-time?      yang:date-and-time
     |  +--ro last-upload-time?          yang:date-and-time
     +--ro supportlink-history* []
        +--ro index?           uint16
        +--ro time-stamp?      yang:date-and-time
        +--ro server?          inet:host
        +--ro port?            inet:port-number
        +--ro error-code?      int32
        +--ro error-message?   string
```

Examples

This section provides basic examples for using the brocade-operation-supportlink module.

Enabling Support Link

This example uses a PATCH request to enable Support Link on the specified switch.

Structure

```
PATCH base_URI/running/brocade-supportlink/supportlink-profile/supportlink-enabled/true
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-supportlink/supportlink-profile/supportlink-enabled/true
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Configuring the Support Link

This example uses a PATCH request to configure Support Link on the specified switch.

Structure

```
PATCH base_URI/running/brocade-supportlink/supportlink-profile/
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-supportlink/supportlink-profile/
```

Request Body

```
{
  "supportlink-profile": {
    "server": "bsnconnect.broadcom.com",
    "port": 443,
    "user-name": "bsl_sl_default_user",
    "period": 1,
    "start-time": 1,
    "start-date": "06/01/2019",
    "collection-time": "0:00",
    "retry-time": 0,
    "end-time-period": 12,
    "proxy-server": "mine",
    "proxy-port": 2,
    "proxy-user": "notmineuser",
    "group-tag": ""
    }
}
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Displaying the Support Link Configuration

This example uses a GET request to display the Support Link configuration on the specified switch.

Structure

```
GET base_URI/running/brocade-supportlink/supportlink-profile/
```

URI

```
GET https://10.10.10.10/rest/running/brocade-supportlink/supportlink-profile/
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "supportlink-profile": {
            "supportlink-enabled": true,
            "server": "bsnconnect.broadcom.com",
            "port": 443,
            "user-name": "bsl_sl_default_user",
            "period": 1,
            "start-time": 1,
            "start-date": "06/01/2019",
            "collection-time": "0:00",
            "retry-time": 0,
            "end-time-period": 12,
            "random-start-time": "06/01/2019 3:35",
            "next-collection-time": "2020-02-16T00:00:00+00:00",
            "next-service-start-time": "2020-02-16T03:35:00+00:00",
            "proxy-server": "mine",
            "proxy-port": 2,
            "proxy-user": "notmineuser",
            "group-tag": ""
        }
    }
}
```

Displaying the Support Link History

This example uses a GET request to display the Support Link history on the specified switch.

Structure

GET *base_URI/*r
unning/brocade-supportlink/supportlink-history/

URI

```
GET https://10.10.10.10/rest/running/brocade-supportlink/supportlink-history/
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
```

```
    <supportlink-history>
        <index>1</index>
        <time-stamp>2020-10-15T23:19:01-06:00</time-stamp>
        <server>asc1.brm.bsnlab.broadcom.net</server>
        <port>8030</port>
        <error-code>-7</error-code>
        <error-message>Inventory file open error</error-message>
    </supportlink-history>
    ...
    <supportlink-history>
        <index>5</index>
        <time-stamp>2020-09-15T00:20:01-06:00</time-stamp>
        <server>asc-gateway.brm.bsnlab.broadcom.net</server>
        <port>8030</port>
        <error-code>-7</error-code>
        <error-message>Inventory file open error</error-message>
    </supportlink-history>
</Response>
```

brocade-time

This module provides enables you to view and configure the time zone and clock server.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the `brocade-time.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-time
  +--rw brocade-time
     +--rw time-zone
     |  +--rw name?                 brocade-time-type:ts-timezone-type
     |  +--rw gmt-offset-hours?     int16
     |  +--rw gmt-offset-minutes?   int16
     +--rw clock-server
     |  +--rw ntp-server-address
     |  |  +--rw server-address*   brocade-time-type:ts-ntp-type
     |  +--ro active-server?        brocade-time-type:ts-ntp-type
     |  +--rw ts-auth-spec?         brocade-time-type:ts-authspec-type
     |  +--rw ts-legacy-mode?       boolean
     +--ro ntp-clock-server* [server]
     |  +--ro server    ts-ntp-type
     |  +--ro index?    int32
     +--rw ntp-clock-server-key* [index]
        +--rw index    int32
        +--rw type?    brocade-time-type:ts-key-type
        +--rw key?     string
```

Examples

This section provides examples for the brocade-time module.

Viewing the Time Zone

The following example uses the GET request to display the current time zone configuration for your device.

Structure

```
GET <base_URI>/running/brocade-time/time-zone
```

URI

The following example uses the GET request to display the current time zone configuration for your device.

```
GET https://10.10.10.10/rest/running/brocade-time/time-zone
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <time-zone>
        <name>America/Toronto</name>
    </time-zone>
</Response>
```

Configuring the Time Zone Offset

The following example uses the PATCH request to modify the time zone offset relative to GMT for your device.

Structure

```
PATCH <base_URI>/running/brocade-time/time-zone
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-time/time-zone
```

Request Body

```
<time-zone>
    <gmt-offset-hours>11</gmt-offset-hours>
    <gmt-offset-minutes>30</gmt-offset-minutes>
</time-zone>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring the Time Zone Name

The following example uses the PATCH request to modify the time zone name for your device.

Structure

```
PATCH <base_URI>/running/brocade-time/time-zone
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-time/time-zone
```

Request Body

```
<time-zone>
    <name>America/Denver</name>
</time-zone>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Viewing the Clock Server Configuration

The following example uses the GET request to display the clock server configuration.

Structure

```
GET <base_URI>/running/brocade-time/clock-server
```

URI

```
GET https://10.10.10.10/rest/running/brocade-time/clock-server
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <clock-server>
        <ntp-server-address>
            <server-address>LOCL</server-address>
        </ntp-server-address>
        <active-server>LOCL</active-server>
        <ts-auth-spec>noauth</ts-auth-spec>
        <ts-legacy-mode>true</ts-legacy-mode>
    </clock-server>
</Response>
```

Configuring the Clock Server

The following example uses the PATCH request to configure a new NTP clock server 11.11.11.11.

Structure

```
PATCH <base_URI>/running/brocade-time/clock-server
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-time/clock-server
```

Request Body

```
<clock-server>
   <ntp-server-address>
        <server-address>11.11.11.11</server-address>
   </ntp-server-address>
</clock-server>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

brocade-traffic-optimizer

This module enables you to view all switches in the fabric that are operational.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-traffic-optimizer.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-traffic-optimizer
  +--rw brocade-traffic-optimizer
     +--ro performance-group-profile* [name]
                               {fibrechannel:fibrechannel_traffic_optimizer_platform}?
     |  +--ro name                                  string
     |  +--ro profile-activated?                    boolean
     |  +--ro profile-scheduled-to-activate?        boolean
     |  +--ro delay-days?                           uint32
     |  +--ro delay-minutes?                        uint32
     |  +--ro profile-description?                  string
     |  +--ro performance-groups
     |  |  +--ro identifier*   performance-group-id-type
     |  +--ro configured-speed-protocol-types
     |  |  +--ro configured-speed-protocol-type*   performance-group-speed-protocol-type
     |  +--ro pending-speed-protocol-types
     |     +--ro configured-speed-protocol-type*   performance-group-speed-protocol-type
     +--ro performance-group* [identifier stats-duration io-category]
                                 {fibrechannel:fibrechannel_traffic_optimizer_platform}?
     |  +--ro identifier                         performance-group-id-type
     |  +--ro stats-duration                     enumeration
     |  +--ro io-category                        enumeration
     |  +--ro performance-group-description?     string
     |  +--ro virtual-channels
     |  |  +--ro virtual-channel*   virtual-channel-type
     |  +--ro aggregate-throughput?              yang:gauge64
     |  +--ro aggregate-iops?                    yang:gauge64
     |  +--ro max-io-latency?                    yang:gauge64
     |  +--ro io-errors?                         yang:gauge64
     |  +--ro flow-count?                        yang:gauge64
     |  +--ro aggregate-violations?              yang:gauge64
     +--ro performance-group-flows* []
                               {fibrechannel:fibrechannel_traffic_optimizer_platform}?
     |  +--ro identifier?        performance-group-id-type
     |  +--ro source-pid?        fibrechannel:fcid-hex-string-type
     |  +--ro destination-pid?   fibrechannel:fcid-hex-string-type
     |  +--ro virtual-channel?   virtual-channel-type
     +--ro partial-performance-group* []
                            {fibrechannel:fibrechannel_traffic_optimizer_platform}?
        +--ro name?      performance-group-id-type
        +--ro reason?    string
        +--ro domains
           +--ro domain*   fibrechannel:domain-id-type
```

Examples

This section provides basic examples for using the brocade-traffic-optimizer module.

Viewing All Traffic Optimizer Counters for a Performance Group

This example uses a GET request to view all predefined traffic optimizer counters for a performance group.

Structure

```
GET <base_URI>/running/brocade-traffic-optimizer/performance-group
```

URI

```
GET https://10.10.10.10/rest/running/brocade-traffic-optimizer/performance-group
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

Viewing All Flows for a Specific Traffic Optimizer Performance Group

This example uses a GET request to view a list of all flows for a specific traffic optimizer group.

Structure

```
GET <base_URI>/running/brocade-traffic-optimizer/performance-group/<performance_group_id>
```

URI

```
GET https://10.10.10.10/rest/running/brocade-traffic-optimizer/performance-group/<performance_group_id>
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

brocade-usb

This module enables you to view the files and folders of an attached USB device.

For a list of changes to this module by release, refer to the individual module YANG file.

Module Tree

This is the tree view of the module from the brocade-usb.yang-tree.txt file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-usb
  +--ro brocade-usb
     +--ro usb-file* []
        +--ro path?   operation-types:usb-path-type
        +--ro name?   operation-types:usb-filename-type
        +--ro size?   uint32
        +--ro date?   yang:date-and-time
```

Examples

This section provides basic examples for using the brocade-usb module.

Viewing the Files and Folders on an Attached USB

This example uses a GET request to view the files and folders on an attached USB.

Structure

```
GET <base_URI>/running/brocade-usb/usb-file
```

URI

```
GET https://10.10.10.10/rest/running/brocade-usb/usb-file
```

Request Body

There is no request body.

Response Body

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <usb-file>
    <path>/usb/usbstorage</path>
    <name>FOS-823540</name>
    <date>2020-12-31T06:54:46+00:00</date>
    <size>189980</size>
  </usb-file>
  <usb-file>
    <path>/usb/usbstorage/config</path>
    <name>config_export.txt</name>
    <date>2021-02-15T13:43:14+00:00</date>
    <size>267724</size>
  </usb-file>
```

```
   <usb-file>
     <path>/usb/usbstorage/defects</path>
     <name>hasm_logs</name>
     <date>2021-02-19T06:23:58+00:00</date>
     <size>117488</size>
   </usb-file>
 <Response>
```

Enabling an Attached USB

This example uses a PATCH request to enable an attached USB.

Structure

```
PATCH <base_URI>/running/brocade-chassis/chassis/usb-device-enabled/true|false
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-chassis/chassis/usb-device-enabled/true
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response contains an empty message body, and a “204 No Content” status appears in the header.

To confirm that the USB is enabled, use a GET request on the chassis (*<base_URI>*/running/brocade-chassis/chassis).

Disabling an Attached USB

This example uses a PATCH request to disable an attached USB.

Structure

```
PATCH <base_URI>/running/brocade-chassis/chassis/usb-device-enabled/true|false
```

URI

```
PATCH https://10.10.10.10/rest/running/brocade-chassis/chassis/usb-device-enabled/false
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response contains an empty message body, and a “204 No Content” status appears in the header.

To confirm that the USB is disabled, use a GET request on the chassis (*<base_URI>*/running/brocade-chassis/chassis).

brocade-zone

This module is used for managing Fibre Channel zoning.

For a list of changes to this module by release, refer to the individual module YANG file.

It assumes a knowledge of fabric zoning as performed in Fabric OS. For information on that topic, refer to *Administering Advanced Zoning* in the *Brocade Fabric OS Administration Guide*.

ModuleTree

This is the tree view of the module from the `brocade-zone.yang-tree.txt` file. See YANG Module Overview for YANG node field definitions and possible values, and see Supported Data Types for data type descriptions.

```
module: brocade-zone
  +--rw brocade-zone
     +--rw defined-configuration {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw cfg* [cfg-name]
     |  |  +--rw cfg-name       zoning-name-type
     |  |  +--rw member-zone
     |  |     +--rw zone-name*   zoning-name-type
     |  +--rw zone* [zone-name]
     |  |  +--rw zone-name           zoning-name-type
     |  |  x--rw zone-type?          zone-type-type
     |  |  +--rw zone-type-string?   zone-type-string-type
     |  |  +--rw member-entry
     |  |     +--rw entry-name*             zone-member-type
     |  |     +--rw principal-entry-name*   zone-member-type
     |  +--rw alias* [alias-name]
     |     +--rw alias-name      zoning-name-type
     |     +--rw member-entry
     |        +--rw alias-entry-name*   union
     +--rw effective-configuration {fibrechannel:fibrechannel_switch_platform}?
     |  +--rw cfg-name?                    zoning-name-type
     |  +--rw checksum?                    string
     |  x--rw cfg-action?                  uint8
     |  +--rw cfg-action-v2?               cfg-action-type
     |  x--rw default-zone-access?         uint8
     |  +--rw default-zone-access-v2?      boolean
     |  +--ro db-max?                      uint32
     |  +--ro db-avail?                    uint32
     |  +--ro db-committed?                uint32
     |  +--ro db-transaction?              uint32
     |  +--ro transaction-token?           uint32
     |  +--ro db-chassis-wide-committed?   uint32
     |  +--ro db-chassis-wide-max?         uint32
     |  +--ro db-fabric-wide-max?          uint32
     |  +--ro domain-with-lowest-db-max?   fibrechannel:domain-id-type
     |  +--ro enabled-zone* [zone-name]
     |     +--ro zone-name           zoning-name-type
     |     x--ro zone-type?          zone-type-type
     |     +--ro zone-type-string?   zone-type-string-type
     |     +--ro member-entry
     |        +--ro entry-name*             union
```

```
|        +--ro principal-entry-name*   union
+--ro fabric-lock {fibrechannel:fibrechannel_switch_platform}?
   +--ro lock-principal-domain-id?           fibrechannel:domain-id-type
   +--ro lock-principal-transaction-token?   uint32
   +--ro timeout?                            uint16
   +--ro remaining-time?                     uint16
   +--ro client-role?                        brocade-security-type:user-config-role-type
   +--ro client-user-name?                   union
   +--ro client-ip-address?                  union
   +--ro client-interface?                   fibrechannel:interface-type
   +--ro client-application-name?            string
```

Examples

This section provides a few examples for using the brocade-zone module.

Retrieving the List of Zone Configurations

This example uses a GET request to retrieve a list of all zone configurations.

Structure

```
GET <base_URI>/running/brocade-zone/defined-configuration/cfg
```

URI

```
GET http://10.10.10.10/rest/running/brocade-zone/defined-configuration/cfg
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
   <cfg>
      <cfg-name>cfg_push</cfg-name>
      <member-zone>
         <zone-name>qosh_112</zone-name>
         <zone-name>qosl_113</zone-name>
         <zone-name>qosm_114</zone-name>
         <zone-name>qosh_115</zone-name>
         <zone-name>qosl_116</zone-name>
         <zone-name>qosm_117</zone-name>
         <zone-name>qosh_ed</zone-name>
      </member-zone>
   </cfg>
   <cfg>
      <cfg-name>cfg_push_165</cfg-name>
      <member-zone>
```

```
            <zone-name>qosh_0</zone-name>
            <zone-name>qosh_3</zone-name>
            <zone-name>qosl_4</zone-name>
            <zone-name>qosl_1</zone-name>
            <zone-name>qosm_2</zone-name>
            <zone-name>qosm_5</zone-name>
         </member-zone>
      </cfg>
      <cfg>
         <cfg-name>ck</cfg-name>
         <member-zone>
            <zone-name>zk</zone-name>
            <zone-name>tdg</zone-name>
            <zone-name>fcoe</zone-name>
         </member-zone>
      </cfg>
      <cfg>
         <cfg-name>test_cfg</cfg-name>
         <member-zone>
            <zone-name>testzone</zone-name>
         </member-zone>
      </cfg>
   </Response>
```

Retrieving the List of Member Zones

This example uses a GET request to retrieve the list of member zones contained in the zone configuration cfg_push.

Structure

```
GET <base_URI>/running/brocade-zone/defined-configuration/cfg/cfg-name/{cfg-name}/member-zone/
```

URI

```
GET http://10.10.10.10/rest/running/brocade-zone/defined-configuration/cfg/cfg-name/cfg_push/member-zone/
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
   <cfg>
      <cfg-name>cfg_push</cfg-name>
      <member-zone>
         <zone-name>qosh_112</zone-name>
         <zone-name>qosl_113</zone-name>
         <zone-name>qosm_114</zone-name>
         <zone-name>qosh_115</zone-name>
```

```
            <zone-name>qosl_116</zone-name>
            <zone-name>qosm_117</zone-name>
            <zone-name>qosh_ed</zone-name>
         </member-zone>
      </cfg>
   </Response>
```

Retrieving a List of the Zone Members in a Specific Zone

This example uses a GET request to retrieve the list of zone members for the qos zone.

Structure

```
GET <base_URI>/running/brocade-zone/defined-configuration/zone/zone-name/{zone.name}
```

URI

```
GET http://10.10.10.10/rest/running/brocade-zone/defined-configuration/zone/zone-name/qos
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers. There are four member entries in this zone.

```
<?xml version="1.0"?>
<Response>
   <zone>
      <zone-name>qos</zone-name>
      <zone-type>1</zone-type>
      <zone-type-string>zone</zone-type-string>
      <member-entry>
         <principal-entry-name>10:10:10:27:f8:8f:44:cd</principal-entry-name>
         <entry-name>10:10:10:27:f8:f0:2a:e8</entry-name>
         <entry-name>10:10:10:27:f8:f0:3a:70</entry-name>
         <entry-name>10:10:10:27:f8:f0:38:70</entry-name>
      </member-entry>
   </zone>
</Response>
```

Adding a Zone to the Defined Configuration

This example uses a POST request to create a user peer zone named “qos” to the defined configuration. This zone has four members.

Structure

```
POST <base_URI>/running/brocade-zone/defined-configuration/zone/zone-name/{zone.name}
```

URI

```
POST http://10.10.10.10/rest/running/brocade-zone/defined-configuration/zone/zone-name/{qos}
```

Parameter

To view the brocade-zone parameters, see brocade-zone/defined-configuration.

To view the brocade-zone specific data types, see Definitions.

Request Body

```
<zone-type-string>user-created-peer-zone</zone-type-string>
<member-entry>
   <principal-entry-name>10:10:10:27:f8:8f:44:cd</principal-entry-name>
   <entry-name>10:10:10:27:f8:f0:2a:e8</entry-name>
   <entry-name>10:10:10:27:f8:f0:3a:70</entry-name>
   <entry-name>10:10:10:27:f8:f0:38:65</entry-name>
</member-entry>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status in the header.

Deleting a Zone from the Defined Configuration

This example uses a DELETE request to delete the zone “zone_host_10” from the “cfg_auto” configuration.

Structure

```
POST <base_URI>/running/brocade-zone/defined-configuration/cfg/cfg-name/{configuration.name}
```

URI

```
POST http://10.10.10.10/rest/running/brocade-zone/defined-configuration/cfg/cfg-name/{cfg_auto}
```

Request Body

```
<member-zone>
    <zone-name>zone_host_10</zone-name>
</member-zone>
```

Response Body

When the operation is successful, the response contains an empty message body and a “200 OK” status in the header.

Creating an Alias

This example uses a POST request to create the alias “ali_1”.

Structure

```
POST <base_URI>/running/brocade-zone/defined-configuration/alias
```

URI

```
POST http://10.10.10.10/rest/running/brocade-zone/defined-configuration/alias
```

Request Body

```
<alias>
   <alias-name>ali_1</alias-name>
   <member-entry>
      <alias-entry-name>1,3</alias-entry-name>
      <alias-entry-name>10:10:10:38:85:9a:13:15</alias-entry-name>
   </member-entry>
</alias>
```

Response Body

When the operation is successful, the response contains an empty message body and a “201 Created” status in the header.

Viewing the Fabric Lock Data

This example uses a GET request to view the fabric lock data.

Structure

```
GET <base_URI>/running/brocade-zone/fabric-lock
```

URI

```
GET http://10.10.10.10/rest/running/brocade-zone/fabric-lock
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers. There are four member entries in this zone.

```
<?xml version="1.0"?>
<Response>
    <fabric-lock>
        <lock-principal-domain-id>0</lock-principal-domain-id>
    </fabric-lock>
</Response>
```

Sample Use Cases

This section provides procedures for using the FOS REST API.

Logging On, Retrieving Switch Information, and Logging Out

This example uses a POST request to log on to a fabric, uses a GET request to retrieve the switch information for a switch, and uses a POST request to log out of the fabric.

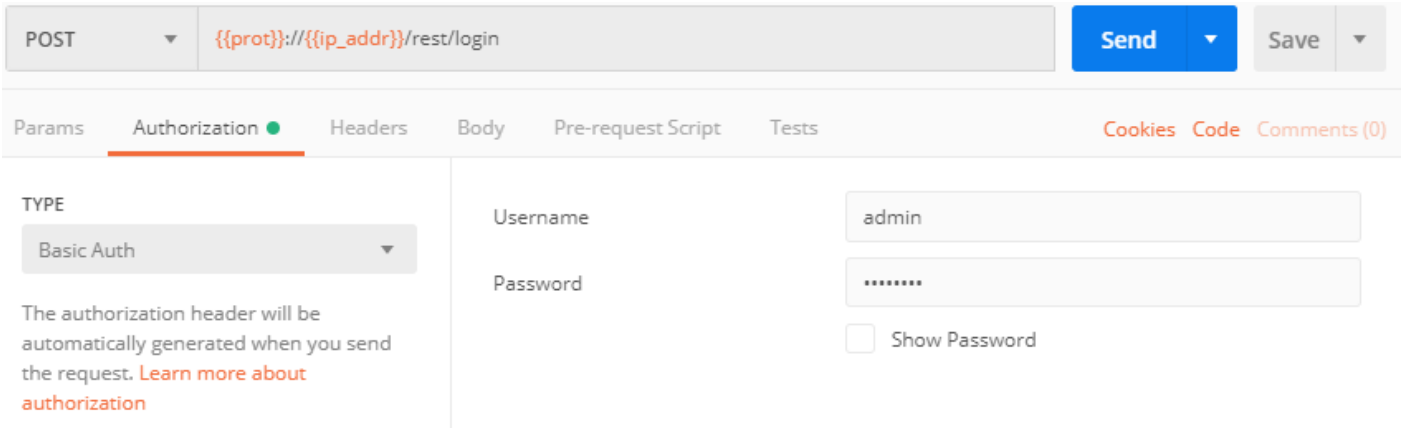
1. Use a POST request to log on to a fabric.

   ```
   POST <base_URI>/login
   ```

   Authorization

   You must provide a valid Fabric OS user name and password (such as, administrator / password) through authorization.

   Figure 4: Authorization

   URI

   ```
   POST https://10.10.10.10/rest/login
   ```

   Request Body

   There is no request body.

   Response Body

   There is no request response. When the operation is successful, the response body contains an empty message body and a “200 OK” status in the header.

   If authentication is successful, a session authorization key is returned to the client in the response headers (for example, Custom_Basic Tk0ZmY2Zjg3NTY2ZDYwYjhmNj5NGQ0NTkzZjM0M2ZlMWM=). Subsequent FOS REST API operations must include this authorization key in the request authorization header. The client applications use this token to obtain further access to the server using the persistent connection.

2. Use a GET request to retrieve the information for a switch.

   ```
   GET <base_URI>running/brocade-fabric/fabric-switch
   ```

   URI

```
GET https://10.10.10.10/rest/running/brocade-fabric/fabric-switch
```

Request Body

No request body is required; however, you must include the session authorization key in the request authorization header.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
   <fabric-switch>
      <name>10:10:14:15:1c:9e:3b:c8</name>
      <chassis-wwn>10:10:14:15:1c:9e:3c:07</chassis-wwn>
      <domain-id>1</domain-id>
      <fcid>16776199</fcid>
      <switch-user-friendly-name>G610_81</switch-user-friendly-name>
      <chassis-user-friendly-name>BrocadeG610</chassis-user-friendly-name>
      <firmware-version>v820</firmware-version>
      <ip-address>10.10.10.1</ip-address>
      <fcip-address>0.0.0.0</fcip-address>
      <ipv6-address>::</ipv6-address>
      <principal>1</principal>
    </fabric-switch>
    <fabric-switch>
      <name>10:10:14:15:1c:a2:1f:40</name>
      <chassis-wwn>10:10:14:15:1c:a2:1f:7f</chassis-wwn>
      <domain-id>3</domain-id>
      <fcid>16776195</fcid>
      <switch-user-friendly-name>G610_82</switch-user-friendly-name>
      <chassis-user-friendly-name>BrocadeG610</chassis-user-friendly-name>
      <firmware-version>v820</firmware-version>
      <ip-address>10.10.10.2</ip-address>
      <fcip-address>0.0.0.0</fcip-address>
      <ipv6-address>::</ipv6-address>
      <principal>0</principal>
    </fabric-switch>
</Response>
```

3. Use a POST request to log out of a fabric.

```
<base_URI>/logout
```

URI

```
POST https://10.10.10.10/rest/logout
```

Request Body

No request body is required; however, you must include the session authorization key in the request authorization header.

Response Body

There is no request response. When the operation is successful, the response contains an empty message body and a “204 No Content” status in the header.

Logging On, Retrieving Chassis Information, and Logging Out Using Session-Less REST API Operation

FOS REST API session-less operation requests complete the login, GET operation, and logout as one complete request. You can use only basic authentication formats for FOS REST API session-less operation, which includes plain text or Base64.

NOTE

- Fabric OS 8.2.2 and later support FOS REST API session-less operation for GET requests only.
- You must have at least one REST API session available to execute session-less requests.

This example uses a GET request with plain text authentication to log on, retrieve information about the chassis in JSON format, and log out using session-less REST API operation.

Use a GET request to obtain chassis information.

Request

```
curl -v -X GET -u admin:password1 -H "Accept: application/yang-data+json" "https://10.11.12.13/rest/running/
brocade-chassis/chassis"
```

Response Body


```
{
 "Response": {
  "chassis": {
   "chassis-user-friendly-name": "sanchassis1",
   "license-id": "10:00:c1:f2:3c:00:af:40",
   "chassis-wwn": "10:00:c1:f2:3c:00:af:4f",
   "serial-number": "FER0304N00Z",
   "manufacturer": "Brocade Communications Systems LLC",
   "part-number": "40-1001199-03",
   "vf-enabled": true,
   "vf-supported": true,
   "fcr-enabled": true,
   "fcr-supported": true,
   "max-blades-supported": 1,
   "vendor-revision-number": "",
   "vendor-part-number": "",
   "vendor-serial-number": "",
   "product-name": "g630",
   "date": "12/24/2020-10:34:37"
  }
 }
```


Creating a Port Trunk Area Using JSON

This example provides step-by-step instructions for creating a port trunk area using JSON. Note that JSON requires that you define additional keys (Accept and Content-Type) in the request headers.

1. Use a GET request to confirm that all ports to be members of the port trunk area are disabled.

   Request Headers

- Authorization = user name and password (such as, administrator / password)
- Accept = application/yang-data+json
- Content-Type = application/yang-data+json

URI

```
GET https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/0%2f16/enabled-state
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "fibrechannel": {
            "name": "0/16",
            "enabled-state": 6
        }
    }
}
```

2. Use a PATCH request to disable that port, if necessary

Request Headers

- Authorization = user name and password (such as, administrator / password)
- Accept = application/yang-data+json
- Content-Type = application/yang-data+json

URI

```
PATCH https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/0%2f16/enabled-state/6
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

3. Use a GET request to confirm that the ports all have trunking enabled.

Request Headers

- Authorization = user name and password (such as, administrator / password)
- Accept = application/yang-data+json
- Content-Type = application/yang-data+json

URI

```
GET https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/0%2f16/trunk-port-enabled
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "fibrechannel": {
            "name": "0/16",
            "trunk-port-enabled": 1
        }
    }
}
```

4. Use a PATCH request to enable trunking on a port, if necessary.

Request Headers

- Authorization = user name and password (such as, administrator / password)
- Accept = application/yang-data+json
- Content-Type = application/yang-data+json

URI

```
PATCH https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/0%2f16/trunk-port-enabled/1
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

5. Use a POST request to create the port trunk area.

Request Headers

- Authorization = user name and password (such as, administrator / password)
- Accept = application/yang-data+json
- Content-Type = application/yang-data+json

URI

```
POST https://10.10.10.10/rest/brocade-fibrechannel-trunk/trunk-area/trunk-index/0
```

Request Body

```
{
    "trunk-members": {
        "trunk-member": [
            "0/16",
            "0/17",
            "0/18",
            "0/19"
        ]
```

```
        }
    }
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status in the headers.

6. Use a PATCH request to enable all ports in the port trunk area.

Request Headers

- Authorization = user name and password (such as, administrator / password)
- Accept = application/yang-data+json
- Content-Type = application/yang-data+json

URI

```
PATCH https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/0%2f16/enabled-state/2
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

7. Use a GET request to confirm the port trunk area creation.

Request Headers

- Authorization = user name and password (such as, administrator / password)
- Accept = application/yang-data+json
- Content-Type = application/yang-data+json

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-trunk/trunk-area/trunk-index/16
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "trunk-area": {
            "trunk-index": 16,
            "trunk-active": true,
            "master-port": "0/16",
            "trunk-members": {
                "trunk-member": [
                    "0/16",
                    "0/17",
                    "0/18",
                    "0/19"
```

```
                    ]
                }
            }
        }
    }
```

Disabling and Enabling a Port

This example uses PATCH to disable and enable a specific port.

The following example first uses GET to identify the state of a gigabitethernet port, uses PATCH to disable the port, uses GET to verify that the state value got changed, and finally uses PATCH to re-enable the port.

1. Use a GET request to identify the enabled-state attribute of the gigabitethernet port 0/3.

   ```
   GET <base_URI>/running/brocade-interface/gigabitethernet/name/0%2f3/enabled-state
   ```

   Request Body

   No request body is required.

   Response Body

   ```
   <?xml version="1.0"?>
   <Response>
     <gigabitethernet>
       <name>0/0</name>
       <enabled-state>1</enabled-state> <-Port is in enabled state.
     </gigabitethernet>
   </Response>
   ```

2. Use a PATCH request to modify the value for enabled-state attribute.

   ```
   PATCH <base_URI>/running/brocade-interface/gigabitethernet/name/0%2f3/enabled-state/0
   ```

   Request Body

   No request body is required.

   Response Body

   When the port status is successfully changed, the response contains an empty message body and a "204 No Content" status in the first line of the headers.

3. Use a GET request to retrieve and verify that the enabled-state attribute value is changed.

   ```
   GET <base_URI>/running/brocade-interface/gigabitethernet/name/0%2f3/enabled-state
   ```

   Request Body

   No request body is required.

   Response Body

   ```
   <?xml version="1.0"?>
   <Response>
     <gigabitethernet>
       <name>0/0</name>
       <enabled-state>0</enabled-state>  <-Port is in disabled state.
     </gigabitethernet>
   </Response>
   ```

4. Use a PATCH request to re-enable the port.

```
PATCH <base_URI>/running/brocade-interface/gigabitethernet/name/0%2f0/enabled-state/1
```

Request Body

No request body is required.

Response Body

When the port status is successfully changed, the response contains an empty message body and a "204 No Content" status in the first line of the headers.

Creating an Up State Tunnel Using Multiple URIs

This example uses POST to create a specific port (IP interface), and then adds an Extension tunnel and a circuit to that port.

1. Make a POST request to create the IP interface on port 4/0.

URI

```
POST <base_URI>/running/brocade-interface/extension-ip-interface
```

Request Body

```
<extension-ip-interface>
  <name>4/0</name>
  <dp-id>1</dp-id>
  <ip-address>10.10.0.10</ip-address>
  <ip-prefix-length>24</ip-prefix-length>
  <mtu-size>1500</mtu-size>
</extension-ip-interface>
```

Response Body

When the interface is successfully created, the response contains an empty message body and a "201 Created" status in the first line of the headers.

NOTE
This is the equivalent of `portcfg ipif 4/ge0.dp1 create 10.10.0.10/24`.

2. Make a GET request to confirm that the IP interface has been created.

URI

```
GET <base_URI>/running/brocade-interface/extension-ip-interface/name/4%2f0/dp-id/1/
```

Request Body

There is no request body.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
  <extension-ip-interface>
    <name>4/0</name>
    <dp-id>1</dp-id>
```

```
    <ip-address>10.10.10.10</ip-address>
    <ip-prefix-length>24</ip-prefix-length>
    <vlan-id>100</vlan-id>
    <mtu-size>1500</mtu-size>
    <status-flags>U R M </status-flags>
  </extension-ip-interface>
</Response>
```

3. Make a POST request to create the tunnel on port 4/30.

   URI

   `POST <base_URI>/running/brocade-extension-tunnel/extension-tunnel`

   Request Body

```
<extension-tunnel>
      <name>4/30</name>
</extension-tunnel>
```

   Response Body

   When the tunnel is successfully created, the response contains an empty message body and a "201 Created" status in the first line of the headers.

   NOTE
   This is the equivalent of `portcfg fciptunnel 4/30 create`.

4. Make a GET request to confirm that the tunnel has been created.

   URI

   `GET <base_URI>/running/brocade-extension-tunnel/extension-circuit-statistics`

   Request Body

   There is no request body.

   Response Body

   When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
  <extension-circuit-statistics>
    <name>4/16</name>
    <circuit-id>1</circuit-id>
    <flow-status>0</flow-status>
    <operational-status>2</operational-status>
    <connection-count>1</connection-count>
    <duration>4450</duration>
    <out-packet-total>926418</out-packet-total>
    <out-packet-lost>0</out-packet-lost>
  </extension-circuit-statistics>
...
  <extension-circuit-statistics>
```

```
      <name>4/30</name>
      <circuit-id>4</circuit-id>
      <flow-status>1</flow-status>
      <operational-status>2</operational-status>
      <connection-count>0</connection-count>
      <duration>0</duration>
      <out-packet-total>0</out-packet-total>
      <out-packet-lost>0</out-packet-lost>
    </extension-circuit-statistics>
  </Response>
```

5. Make a POST request to create circuit 0.

   URI

   `POST <base_URI>/running/brocade-extension-tunnel/extension-circuit`

   Request Body

```
<extension-circuit>
  <name>4/30</name>
  <circuit-id>0</circuit-id>
  <local-ip-address>10.10.0.10</local-ip-address>
  <remote-ip-address>10.10.0.20</remote-ip-address>
  <minimum-communication-rate>1000000</minimum-communication-rate>
  <maximum-communication-rate>1000000</maximum-communication-rate>
</extension-circuit>
```

   Response Body

   When the circuit is successfully created, the response contains an empty message body and a "201 Created" status in the first line of the headers.

   NOTE
   This is the equivalent of `portcfg fcipcircuit 4/30 create 0 --local-ip 10.10.0.10 --remote-ip 10.10.0.20 --min-comm-rate 1000000 --max-comm-rate 1000000`.

6. Make a GET request to confirm that the tunnel has been created.

   URI

   `GET <base_URI>/running/brocade-extension-tunnel/extension-circuit-statistics`

   Request Body

   There is no request body.

   Response Body

   When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
  <extension-circuit-statistics>
    <name>4/16</name>
    <circuit-id>1</circuit-id>
    <flow-status>0</flow-status>
```

```
      <operational-status>2</operational-status>
      <connection-count>1</connection-count>
      <duration>4450</duration>
      <out-packet-total>926418</out-packet-total>
      <out-packet-lost>0</out-packet-lost>
    </extension-circuit-statistics>
...
    <extension-circuit-statistics>
      <name>4/30</name>
      <circuit-id>4</circuit-id>
      <flow-status>1</flow-status>
      <operational-status>2</operational-status>
      <connection-count>0</connection-count>
      <duration>0</duration>
      <out-packet-total>0</out-packet-total>
      <out-packet-lost>0</out-packet-lost>
    </extension-circuit-statistics>
</Response>
```

Running a ClearLink Diagnostic Port Test

This example demonstrates using the FOS REST API to perform a ClearLink diagnostic test on an existing D_Port.

1. Use a GET request to verify that the port (0/89) is configured as a diagnostic port (D_Port). Notice that the slash character in the name resource is encoded as “%2f”.

   URI

   ```
   GET <base_URI>/brocade-interface/fibrechannel/name/0%2f89/d-port-enable
   ```

   Request Body

   No request body is required.

   Response Body

   When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

   ```
   <?xml version="1.0"?>
   <Response>
       <fibrechannel>
           <name>0/89</name>
           <d-port-enable>1</d-port-enable>
       </fibrechannel>
   </Response>
   ```

   Per the YANG definition, a “1” in the “d-port-enable” field indicates that the port is configured as a diagnostic port.

2. Use a PATCH request to start the ClearLink diagnostic test on the desired D_Port. This example uses a diagnostic test frame-size value of 2048.

   URI

   ```
   PATCH <base_URI>/brocade-diagnostics/fibrechannel-diagnostics/name/0%2f89/diagnostic-control/1
   ```

   Request Body

   ```
   <frame-size>2048</frame-size>
   ```

Response Body

When the operation is successful, the response has a message body similar to the following, and a “204 No Content” status in the headers.

3. Use a GET request to retrieve the diagnostic results.

URI

```
GET <base_URI>/brocade-diagnostics/fibrechannel-diagnostics/name/0%2f89
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-diagnostics>
        <name>0/89</name>
        <diagnostic-control>0</diagnostic-control>
        <mode>automatic</mode>
        <state>STOPPED</state>
        <distance>-1</distance>
        <electrical-loopback-test>
            <result>NOT STARTED</result>
            <comments>----------</comments>
        </electrical-loopback-test>
        <optical-loopback-test>
            <result>NOT STARTED</result>
            <comments>----------</comments>
        </optical-loopback-test>
        <link-traffic-test>
            <result>NOT STARTED</result>
            <comments>----------</comments>
        </link-traffic-test>
        <start-time>Fri Aug 24 09:39:18 2018</start-time>
        <frame-count>1</frame-count>
        <frame-size>1024</frame-size>
        <time/>
        <fec>
            <enable>yes</enable>
            <active>yes</active>
            <option>no</option>
        </fec>
        <cr>
            <enable>yes</enable>
            <active>no</active>
            <option>no</option>
        </cr>
        <rt-latency>0</rt-latency>
        <buffers-required/>
```

```
            <failure-report>
                <errors-detected-local>no</errors-detected-local>
                <errors-detected-remote>no</errors-detected-remote>
            </failure-report>
            <end-time>Fri Aug 24 09:39:39 2018</end-time>
            <egress-power-loss>
                <tx>-5.57</tx>
                <rx/>
                <loss/>
                <comments/>
            </egress-power-loss>
            <ingress-power-loss>
                <tx/>
                <rx>-1.28</rx>
                <loss/>
                <comments/>
            </ingress-power-loss>
            <payload-pattern>
                <pattern>jCRPAT</pattern>
            </payload-pattern>
        </fibrechannel-diagnostics>
    </Response>
```

Creating a MAPS Rule to Monitor CRC Errors on FC Ports

This example demonstrates using the FOS REST API to create a MAPS rule to monitor CRC errors on FC ports.

1. Use a POST request to create a MAPS rule to monitor CRC errors on all the ports with a RASlog alert action.

   URI

   ```
   POST <base_URI>/brocade-maps/rule
   ```

   Request Body

   ```
   <rule>
       <name>rule_to_monitor_crc_error</name>
       <is-rule-on-rule>false</is-rule-on-rule>
       <monitoring-system>CRC</monitoring-system>
       <time-base>MIN</time-base>
       <logical-operator>g</logical-operator>
       <threshold-value>10</threshold-value>
       <group-name>ALL_PORTS</group-name>
       <actions>
           <action>raslog</action>
       </actions>
   </rule>
   ```

   Response Body

   When the operation is successful, the response has an empty message body and a “201 Created” status in the headers.

2. Use a GET request to verify the newly created rule and its contents.

   URI

```
GET <base_URI>/brocade-maps/rule/name/rule_to_monitor_crc_error
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <rule>
        <name>rule_to_monitor_crc_error</name>
        <is-rule-on-rule>false</is-rule-on-rule>
        <monitoring-system>CRC</monitoring-system>
        <time-base>MIN</time-base>
        <logical-operator>g</logical-operator>
        <threshold-value>0</threshold-value>
        <group-name>ALL_PORTS</group-name>
        <actions>
            <action>raslog</action>
        </actions>
    </rule>
</Response>
```

3. Use a POST request to create a policy with the newly created rule and activate the policy.

URI

```
POST <base_URI>/brocade-maps/maps-policy
```

Request Body

```
<maps-policy>
    <name>monitor_crc_error</name>
    <rule-list>
        <rule>rule_to_monitor_crc_error</rule>
    </rule-list>
    <is-active-policy>true</is-active-policy>
</maps-policy>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status in the headers.

4. Use a PATCH request to enable the RASlog action at switch level.

URI

```
PATCH <base_URI>/brocade-maps/maps-config
```

Request Body

No request body is required.

```
<maps-config>
    <actions>
        <action>raslog</action>
    </actions>
</maps-config>
```

Response Body

When the operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

Monitoring the Execution of a MAPS Rule

This example demonstrates using the FOS REST API to monitor the execution of a MAPS rule.

1. Use a GET request to retrieve the dashboard data for executed rules.

   URI

   ```
   GET <base_URI>/brocade-maps/dashboard-rule
   ```

   Request Body

   No request body is required.

   Response Body

   When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

   ```
   <?xml version="1.0"?>
   <Response>
       <dashboard-rule>
           <category>Port Health</category>
           <triggered-count>42</triggered-count>
           <name>rule_to_monitor_crc_error</name>
           <time-stamp>Aug 04 21:50:54</time-stamp>
           <repetition-count>1</repetition-count>
           <objects>
               <object>F-Port 10:90</object>
           </objects>
       </dashboard-rule>
   </Response>
   ```

2. Verify the RASLOG alert.

   ```
   2019/08/04-21:50:54, [MAPS-1003], 10737, FID 128, WARNING, TOM_2_____________________TOM_2, port10, F-Port
    10, Condition=ALL_PORTS(CRC/min>0), Current Value:[CRC, 90 CRCs], RuleName=rule_to_monitor_crc_error,
    Dashboard Category=Port Health.
   ```

Creating a User-Defined Group, Adding Ports to the Group, and Using the Group to Monitor a Rule

This example demonstrates using the FOS REST API to create a user-defined group, add ports to the group, and use the group to monitor a rule.

1. Use a POST request to create a user-defined port group.

   URI

   ```
   POST <base_URI>/brocade-maps/group
   ```

   Request Body

   ```
   <group>
       <name>new_group_to_mon_ITW</name>
       <group-type>fc-port</group-type>
   </group>
   ```

   Response Body

   When the operation is successful, the response has an empty message body and a “201 Created” status in the headers.

2. Use a POST request to add ports to the group.

   URI

   ```
   POST <base_URI>/brocade-maps/rule
   ```

   Request Body

   ```
   <group>
       <name>new_group_to_mon_ITW</name>
       <members>
           <member>0/12</member>
       </members>
   </group>
   ```

   Response Body

   When the operation is successful, the response has an empty message body and a “201 Created” status in the headers.

3. Use a POST request to create a new rule to monitor ITW errors on the ports.

   URI

   ```
   POST <base_URI>/brocade-maps/rule
   ```

   Request Body

   ```
   <rule>
       <name>rule_to_monitor_ITW_error</name>
       <is-rule-on-rule>false</is-rule-on-rule>
       <monitoring-system>CRC</monitoring-system>
       <time-base>MIN</time-base>
       <logical-operator>g</logical-operator>
   ```

```
        <threshold-value>10</threshold-value>
        <group-name>new_group_to_mon_ITW</group-name>
        <actions>
            <action>raslog</action>
        </actions>
    </rule>
```

Response Body

When the operation is successful, the response has an empty message body and a “201 Created” status in the headers.

Retrieving the Switch Status Policy Report

This example demonstrates using the FOS REST API to retrieve the switch status policy report.

Use a GET request to retrieve the switch status policy report.

URI

```
GET <base_URI>/brocade-maps/switch-status-policy-report
```

Request Body

No request body is required.

Response Body

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <switch-status-policy-report>
        <switch-health>healthy</switch-health>
        <power-supply-health>healthy</power-supply-health>
        <fan-health>healthy</fan-health>
        <wwn-health>healthy</wwn-health>
        <temperature-sensor-health>healthy</temperature-sensor-health>
        <ha-health>healthy</ha-health>
        <control-processor-health>healthy</control-processor-health>
        <core-blade-health>healthy</core-blade-health>
        <blade-health>healthy</blade-health>
        <flash-health>healthy</flash-health>
        <marginal-port-health>healthy</marginal-port-health>
        <faulty-port-health>healthy</faulty-port-health>
        <missing-sfp-health>healthy</missing-sfp-health>
        <error-port-health>healthy</error-port-health>
        <expired-certificate-health>healthy</expired-certificate-health>
        <airflow-mismatch-health>healthy</airflow-mismatch-health>
        <trusted-fos-cert-health>healthy</trusted-fos-cert-health>
    </switch-status-policy-report>
</Response>
```

Generating a CSR and Importing a Security Certificate

The following example first uses a POST request to generate a CSR (certificate signing request), uses a PATCH request to export the CSR to third-party CA (certificate authority), uses a PATCH request to import the client CA on the local switch, and finally uses a PATCH request to import the signed switch certificate.

1. Use a POST request to generate a CSR (certificate signing request).

   URI

   ```
   POST https://10.10.10.10/rest/running/brocade-security/security-certificate-generate
   ```

   Request Body

   ```
   <security-certificate-generate>
       <certificate-entity>csr</certificate-entity>
       <certificate-type>fcap</certificate-type>
       <algorithm-type>rsa</algorithm-type>
       <key-size>2048</key-size>
       <hash-type>sha256</hash-type>
       <years>10</years>
       <country-name>US</country-name>
       <state-name>Colorado</state-name>
       <locality-name>Smallville</locality-name>
       <organization-name>Acme Inc.</organization-name>
       <unit-name>abc_division</unit-name>
       <domain-name>10.10.10.10.abc.acme.com</domain-name>
   </security-certificate-generate>
   ```

   Response Body

   When the operation is successful, the response has an empty message body and a “201 Created” status message.

   NOTE
   Generating a CSR for HTTPS may restart HTTP and HTTPS and inadvertently terminate REST operations.

2. Use a PATCH request to export a CSR to third-party CA (certificate authority). The CA then signs the CSR and generates a switch certificate.

   URI

   ```
   PATCH https://10.10.10.10/rest/running/brocade-security/security-certificate-action
   ```

   Request Body

   ```
   <security-certificate-action>
       <certificate-entity></certificate-entity>
       <certificate-name></certificate-name>
       <certificate-type></certificate-type>
       <operation></operation>
       <protocol></protocol>
       <remote-directory></remote-directory>
       <remote-host-ip></remote-host-ip>
       <remote-user-name></remote-user-name>
       <remote-user-password></remote-user-password>
   </security-certificate-action>
   ```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

3. Use a PATCH request to import the client CA on the local switch, which is the CA certificate that signed the switch certificate.

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/security-certificate-action
```

Request Body

```
<security-certificate-action>
    <certificate-entity></certificate-entity>
    <certificate-name></certificate-name>
    <certificate-type></certificate-type>
    <operation></operation>
    <protocol></protocol>
    <remote-directory></remote-directory>
    <remote-host-ip></remote-host-ip>
    <remote-user-name></remote-user-name>
    <remote-user-password></remote-user-password>
</security-certificate-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

4. Use a PATCH request to import the signed switch certificate.

URI

```
PATCH https://10.10.10.10/rest/running/brocade-security/security-certificate-action
```

Request Body

```
<security-certificate-action>
    <protocol>scp</protocol>
    <certificate-entity>cert</certificate-entity>
    <certificate-type>fcap</certificate-type>
    <certificate-name>10.10.10.10.fcap.pem</certificate-name>
    <operation>import</operation>
    <remote-host-ip>10.20.20.20</remote-host-ip>
    <remote-user-name>ca_user</remote-user-name>
    <remote-directory>/certs</remote-directory>
    <remote-user-password>cGFzc3dvcmQxMjM=</remote-user-password>
</security-certificate-action>
```

Response Body

When the operation is successful, the response has an empty message body and a “204 No Content” status message.

Configuring SSH Public Key Authentication on a Switch for Incoming Connections

The following example first uses a POST request to generate and install an 'dsa' private key on the device, then uses a PATCH request to import a public key from a remote host.

1. Use a POST request to generate and install an 'dsa' private key on the device.
   Note that the passphrase used in this example is Base64 encoded.

   URI

   ```
   POST https://10.10.10.10/rest/running/brocade-security/sshutil-key
   ```

   Request Body

   ```
   <sshutil-key>
       <algorithm-type>dsa</algorithm-type>
       <key-type>public-private-key</key-type>
       <passphrase>U2lsYfraW4xMjM=</passphrase>
   </sshutil-key>
   ```

   Response Body

   When the operation is successful, the response has an empty message body and a “201 Created” status message.

2. Use a PATCH request to import a public key from a remote host.
   Note that the password used in this example is Base64 encoded.

   URI

   ```
   PATCH https://10.10.10.10/rest/running/brocade-security/sshutil-public-key-action
   ```

   Request Body

   ```
   <sshutil-public-key-action>
       <action>import</action>
       <algorithm-type>rsa</algorithm-type>
       <user-name>auserlocal</user-name>
       <remote-user-name>auserremote</remote-user-name>
       <remote-host-ip>11.11.11.11</remote-host-ip>
       <remote-user-password>U2lsYfraW4xMjM=</remote-user-password>
       <remote-directory>~auser/.ssh</remote-directory>
       <public-key-name>in_coming.pub</public-key-name>
   </sshutil-public-key-action>
   ```

   Response Body

   When the operation is successful, the response has an empty message body and a “204 No Content” status message.

3. Test the setup by logging in to the switch from a remote device, or by running a command remotely using SSH.

Configuring SSH Public Key Authentication on a Switch for Outgoing Connections

The following example first uses a POST request to generate and install an 'dsa' private key on the device, then uses a PATCH request to export a public key to a remote host.

1. Use a POST request to generate and install an 'dsa' private key on the device.
   Note that the passphrase used in this example is Base64 encoded.

   URI

   ```
   POST https://10.10.10.10/rest/running/brocade-security/sshutil-key
   ```

   Request Body

   ```
   <sshutil-key>
       <algorithm-type>dsa</algorithm-type>
       <key-type>public-private-key</key-type>
       <passphrase>U2lsYfraW4xMjM=</passphrase>
   </sshutil-key>
   ```

   Response Body

   When the operation is successful, the response has an empty message body and a “201 Created” status message.

2. Use a PATCH request to export a public key to a remote host.
   Note that the password used in this example is Base64 encoded.

   URI

   ```
   PATCH https://10.10.10.10/rest/running/brocade-security/sshutil-public-key-action
   ```

   Request Body

   ```
   <sshutil-public-key-action>
        <action>export</action>
        <algorithm-type>rsa</algorithm-type>
        <user-name>auserlocal</user-name>
        <remote-user-name>auserremote</remote-user-name>
        <remote-host-ip>11.11.11.11</remote-host-ip>
        <remote-user-password>U2lsYfraW4xMjM=</remote-user-password>
        <remote-directory>~auser/.ssh</remote-directory>
        <public-key-name>out_going.pub</public-key-name>
   </sshutil-public-key-action>
   ```

   Response Body

   When the operation is successful, the response has an empty message body and a “204 No Content” status message.

3. Append the public key to a remote host by logging in to the remote host, locating the directory where authorized keys are stored, and appending the public key to the file.
   You may need to refer to the host’s documentation to locate where the authorized keys are stored.

4. Test the setup by using a command that uses SCP and authentication.

Creating a New Zone Using REST

In this use case, use a POST request to create a new zone, “zone1” with three zone members “10:00:00:00:00:00:00:01”, “10:00:00:00:00:00:00:02”, and "10:00:00:00:00:00:00:03".

1. Use a POST request to create the new zone, “zone1”.

   URI

   ```
   POST <base_URI>/rest/running/brocade-zone/zone
   ```

   Request Body

   ```
   <zone>
      <zone-name>zone1</zone-name>
      <zone-type-string>zone</zone-type-string>
      <member-entry>
         <entry-name>10:00:00:00:00:00:00:01</entry-name>
         <entry-name>10:00:00:00:00:00:00:02</entry-name>
         <entry-name>10:00:00:00:00:00:00:03</entry-name>
      </member-entry>
   </zone>
   ```

   Response Body

   When the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

2. Use a GET request to verify the new zone, "zone1".

   URI

   ```
   GET <base_URI>/rest/running/brocade-zone/zone/zone-name/zone1
   ```

   Request Body

   No request body is required.

   Response Body

   The response contains a message body similar to the following, and has a “200 OK” status in the first line of the headers.

   ```
   <?xml version="1.0"?>
   <Response>
   <zone>
      <zone-name>zone1</zone-name>
      <zone-type>0</zone-type>
      <zone-type-string>zone</zone-type-string>
      <member-entry>
         <entry-name>10:00:00:00:00:00:00:01</entry-name>
         <entry-name>10:00:00:00:00:00:00:02</entry-name>
         <entry-name>10:00:00:00:00:00:00:03</entry-name>
      </member-entry>
   </zone>
   </Response>
   ```

Adding Additional Zone Members to an Existing Zone

In this use case, "zone1" already exists with zone members “10:00:00:00:00:00:00:01”, “10:00:00:00:00:00:00:02”, and "10:00:00:00:00:00:00:03". This will add new zone members “10:00:00:00:00:00:00:04”, “10:00:00:00:00:00:00:05”, and "10:00:00:00:00:00:00:06" to the existing “zone1”.

Note that you can use a PATCH request to replace all zone members in a zone. You can perform a PATCH request on a cfg, a zone, or an alias.

1. Use a GET request to verify the zone.

   URI

   ```
   GET <base_URI>/rest/running/brocade-zone/zone/zone-name/zone1
   ```

   Request Body

   No request body is required.

   Response Body

   The response contains a message body similar to the following, and has a “200 OK” status in the first line of the headers.

   ```
   <?xml version="1.0"?>
   <Response>
   <zone>
      <zone-name>zone1</zone-name>
      <zone-type>0</zone-type>
      <zone-type-string>zone</zone-type-string>
      <member-entry>
         <entry-name>10:00:00:00:00:00:00:01</entry-name>
         <entry-name>10:00:00:00:00:00:00:02</entry-name>
         <entry-name>10:00:00:00:00:00:00:03</entry-name>
      </member-entry>
   </zone>
   </Response>
   ```

2. Use a POST request to add new zone members “10:00:00:00:00:00:00:04”, “10:00:00:00:00:00:00:05”, and "10:00:00:00:00:00:00:06" to the zone1.

   URI

   ```
   POST <base_URI>/rest/running/brocade-zone/defined-configuration/zone
   ```

   Request Body

   ```
   <zone>
      <zone-name>zone1</zone-name>
      <zone-type-string>zone</zone-type-string>
      <member-entry>
         <entry-name>10:00:00:00:00:00:00:04</entry-name>
         <entry-name>10:00:00:00:00:00:00:05</entry-name>
         <entry-name>10:00:00:00:00:00:00:06</entry-name>
      </member-entry>
   </zone>
   ```

   Response Body

   When the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

Creating, Modifying, and Deleting a Zone Using REST

This use case set walks you through the following actions:

- Creating a zone
- Modifying a zone
- Aborting a zone transaction
- Deleting a zone

In this use case set, zone configuration “cfg1” already exists with member zones “zone1”, “zone2”, and “zone4”. We will create a new zone, “zone3” and add it to the “cfg1” zone configuration. We will then delete “zone4” from the zone configuration.

Creating a New Zone in an Existing Configuration Using REST

In this use case, zone configuration “cfg1” already exists with member zones “zone1”, and “zone2”. This will create a new zone, “zone3”, and add it to the existing “cfg1” zone configuration.
This procedures requires that you record the checksum value for the effective zone. The checksum is a MD5 (Message-Digest algorithm 5) calculation (128-bit cryptographic hash) used to verify data integrity. You need the checksum to save changes to the effective configuration. For more information about the checksum, see brocade-zone/effective-configuration.

1. Use a GET request to verify the current zone configuration.

   URI

   ```
   GET <base_URI>/rest/running/brocade-zone/effective-configuration
   ```

   Request Body

   No request body is required.

   Response Body

   The response contains a message body similar to the following, and has a “200 OK” status in the first line of the headers.

   ```
   <?xml version="1.0"?
   {
       "Response": {
           "effective-configuration": {
               "default-zone-access": 1,
               "default-zone-access-v2": true,
               "db-max": 4194304,
               "db-avail": 1036360,
               "db-committed": 5656,
               "db-transaction": 0,
               "db-chassis-wide-committed": 21715,
               "db-chassis-wide-max": 16777216,
               "db-fabric-wide-max": 1045274,
               "domain-with-lowest-db-max": 92,
               "transaction-token": 0,
               "checksum": "62c211e568f0a264ec7ab4692eac7b65",
               "cfg-name": "default",
               "enabled-zone": [
                   {
                       "zone-name": "68161_EMC",
                       "zone-type": 0,
   ```

```
                    "zone-type-string": "zone",
                    "member-entry": {
                        "entry-name": [
                            "20:00:8c:7c:ff:22:f8:80",
                            "20:00:8c:7c:ff:22:f8:81",
                            "50:00:09:73:58:0d:d8:5c",
                            "50:00:09:73:58:0d:db:ff"
                        ]
                    }
                },
                {
                    "zone-name": "68163_EMC",
                    "zone-type": 0,
                    "zone-type-string": "zone",
                    "member-entry": {
                        "entry-name": [
                            "10:00:00:00:c9:f8:04:35",
                            "10:00:00:00:c9:f8:04:36",
                            "50:00:09:73:58:0d:d8:1c",
                            "50:00:09:73:58:0d:d8:5c"
                        ]
                    }
                },
...
                {
                    "zone-name": "68166_NETAPP3",
                    "zone-type": 0,
                    "zone-type-string": "zone",
                    "member-entry": {
                        "entry-name": [
                            "10:00:00:10:9b:8f:2d:0f",
                            "10:00:00:10:9b:8f:2d:10",
                            "20:83:00:a0:98:6b:08:7c",
                            "20:84:00:a0:98:6b:08:7c",
                            "20:85:00:a0:98:6b:08:7c",
                            "20:86:00:a0:98:6b:08:7c"
                        ]
                    }
                },
                {
                    "zone-name": "lsan_68163_NETAPP",
                    "zone-type": 0,
                    "zone-type-string": "zone",
                    "member-entry": {
                        "entry-name": [
                            "10:00:00:00:c9:f8:04:35",
                            "10:00:00:00:c9:f8:04:36",
                            "20:77:00:a0:98:6b:08:7c",
                            "20:78:00:a0:98:6b:08:7c",
                            "20:79:00:a0:98:6b:08:7c",
                            "20:7a:00:a0:98:6b:08:7c"
                        ]
                    }
```

```
                    }
                ]
            }
        }
    }
```

2. Record the checksum value.
3. Use a POST request to create the new zone, “zone3”.

   URI

   ```
   POST <base_URI>/rest/running/brocade-zone/defined-configuration/zone
   ```

   Request Body

   ```
   <zone>
      <zone-name>zone3</zone-name>
      <zone-type-string>zone</zone-type-string>
      <member-entry>
         <entry-name>10:00:00:00:00:00:00:07</entry-name>
         <entry-name>10:00:00:00:00:00:00:08</entry-name>
         <entry-name>10:00:00:00:00:00:00:09</entry-name>
      </member-entry>
   </zone>
   ```

   Response Body

   When the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

4. Use a PATCH request to add “zone3” to the existing configuration “cfg1” with all desired zone members. Because a PATCH operation overwrites the entire member-zone leaf-list for “cfg1”, all desired member-zones must be included, including any member zones that were already in the configuration.

   Note that you can use a POST request to add a zone (zone3) to an existing configuration (cfg1). You can perform a POST request on a cfg, a zone, or an alias.

   URI

   ```
   PATCH <base_URI>/rest/running/brocade-zone/defined-configuration/cfg
   ```

   Request Body

   ```
   <cfg>
       <cfg-name>cfg1</cfg-name>
       <member-zone>
           <zone-name>zone1</zone-name>
           <zone-name>zone2</zone-name>
           <zone-name>zone3<zone-name>
       </member-zone>
   </cfg>
   ```

   Response Body

   When the resource is successfully created, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

5. Use a PATCH request to save the zone configuration using the zoneDB checksum from step 1.

   URI

```
PATCH <base_URI>/rest/running/brocade-zone/effective-configuration/cfg-action-v2/save
```

Request Body

```
<checksum>fd0d26c940d4f55b467b76552d687e74</checksum>
```

Response Body

When the request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

6. Use a GET request to retrieve the new zoneDB checksum value.

   URI

   ```
   GET <base_URI>/rest/running/brocade-zone/effective-configuration/checksum
   ```

   Request Body

   No request body is required.

   Response Body The response contains a message body similar to the following, and has a “200 OK” status in the first line of the headers.

   ```
   <?xml version="1.0"?>
   <Response>
       <effective-configuration>
           <checksum>d2ba41573853f387d9fd1d76f3b56112</checksum>
       </effective-configuration>
   </Response>
   ```

7. Use a PATCH request to enable the new zone configuration using the zoneDB checksum from step 6.

   URI

   ```
   PATCH <base_URI>/rest/running/brocade-zone/effective-configuration/cfg-name/cfg1
   ```

   Request Body

   ```
   <checksum>d2ba41573853f387d9fd1d76f3b56112</checksum>
   ```

   Response Body

   When the request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

8. Use a GET request to view the new effective zoning configuration.

   URI

   ```
   GET <base_URI>/rest/running/brocade-zone/effective-configuration
   ```

   Request Body

   No request body is required.

   Response Body

   The response contains a message body similar to the following, and has a “200 OK” status in the first line of the headers. The zone named “zone3” has been added to the zoning configuration named “cfg1”.

   ```
   <?xml version="1.0"?>
   {
       "Response": {
           "effective-configuration": {
   ```

```
                "default-zone-access": 1,
                "default-zone-access-v2": true,
                "db-max": 4194304,
                "db-avail": 1036360,
                "db-committed": 5656,
                "db-transaction": 0,
                "db-chassis-wide-committed": 21715,
                "db-chassis-wide-max": 16777216,
                "db-fabric-wide-max": 1045274,
                "domain-with-lowest-db-max": 92,
                "transaction-token": 0,
                "checksum": "62c211e568f0a264ec7ab4692eac7b65",
                "cfg-name": "default",
                "enabled-zone": [
                    {
                        "zone-name": "68161_EMC",
                        "zone-type": 0,
                        "zone-type-string": "zone",
                        "member-entry": {
                            "entry-name": [
                                "20:00:8c:7c:ff:22:f8:80",
                                "20:00:8c:7c:ff:22:f8:81",
                                "50:00:09:73:58:0d:d8:5c",
                                "50:00:09:73:58:0d:db:ff"
                            ]
                        }
                    },
                    {
                        "zone-name": "68163_EMC",
                        "zone-type": 0,
                        "zone-type-string": "zone",
                        "member-entry": {
                            "entry-name": [
                                "10:00:00:00:c9:f8:04:35",
                                "10:00:00:00:c9:f8:04:36",
                                "50:00:09:73:58:0d:d8:1c",
                                "50:00:09:73:58:0d:d8:5c"
                            ]
                        }
                    },
...
                    {
                        "zone-name": "68166_NETAPP3",
                        "zone-type": 0,
                        "zone-type-string": "zone",
                        "member-entry": {
                            "entry-name": [
                                "10:00:00:10:9b:8f:2d:0f",
                                "10:00:00:10:9b:8f:2d:10",
                                "20:83:00:a0:98:6b:08:7c",
                                "20:84:00:a0:98:6b:08:7c",
                                "20:85:00:a0:98:6b:08:7c",
                                "20:86:00:a0:98:6b:08:7c"
```

```
                        ]
                    }
                },
                {
                    "zone-name": "lsan_68163_NETAPP",
                    "zone-type": 0,
                    "zone-type-string": "zone",
                    "member-entry": {
                        "entry-name": [
                            "10:00:00:00:c9:f8:04:35",
                            "10:00:00:00:c9:f8:04:36",
                            "20:77:00:a0:98:6b:08:7c",
                            "20:78:00:a0:98:6b:08:7c",
                            "20:79:00:a0:98:6b:08:7c",
                            "20:7a:00:a0:98:6b:08:7c"
                        ]
                    }
                }
            ]
        }
    }
}
```

Modifying a Zone

This example modifies the existing "zone2" in the defined-configuration “cfg1”, which is currently enabled.

1. Use a GET request to retrieve the current zoneDB checksum value.

   URI

   ```
   GET <base_URI>/rest/running/brocade-zone/effective-configuration/checksum
   ```

   Request Body

   No request body is required.

   Response Body

   The response contains a message body similar to the following, and has a “200 OK” status in the headers.

   ```
   <?xml version="1.0"?>
   <Response>
      <effective-configuration>
          <checksum>fd0d26c940d4f55b467b76552d687e74</checksum> <== record this value
      </effective-configuration>
   </Response>
   ```

2. Use a GET request to verify the current zone “zone2” configuration.

   URI

   ```
   GET <base_URI>/rest/running/brocade-zone/defined-configuration/zone/zone-name/zone2
   ```

   Request Body

   No request body is required.

   Response Body

   The response contains a message body similar to the following, and has a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <zone>
        <zone-name>zone2</zone-name>
        <zone-type>0</zone-type>
        <member-entry>
            <entry-name>10:00:00:00:00:00:00:04</entry-name>
            <entry-name>10:00:00:00:00:00:00:05</entry-name>
            <entry-name>10:00:00:00:00:00:00:06</entry-name>
        </member-entry>
    </zone>
</Response>
```

3. Use a PATCH request to update the existing zone “zone2”. Because a PATCH operation overwrites the entire member-entry leaf-list for “zone2”, all desired member-entries must be included, including any member entries that were already in the configuration.

   URI

   ```
   PATCH <base_URI>/rest/running/brocade-zone/defined-configuration/zone
   ```

   Request Body

   ```
   <zone>
       <zone-name>zone2</zone-name>
       <zone-type>0</zone-type>
       <member-entry>
           <entry-name>10:00:00:00:00:00:00:04</entry-name>
           <entry-name>10:00:00:00:00:00:00:05</entry-name>
           <entry-name>10:00:00:00:00:00:00:06</entry-name>
           <entry-name>10:00:00:00:00:00:00:07</entry-name>
           <entry-name>10:00:00:00:00:00:00:08</entry-name>
           <entry-name>10:00:00:00:00:00:00:09</entry-name>
       </member-entry>
   </zone>
   ```

   Response Body

   When the operation completes successfully, the response returns an empty message body and a “204 No Content” status in the headers.

4. Use a PATCH request to enable the updated zone configuration using the zoneDB checksum from step 1.

   URI

   ```
   PATCH <base_URI>/rest/running/brocade-zone/effective-configuration/cfg-name/cfg1
   ```

   Request Body

   ```
   <checksum>fd0d26c940d4f55b467b76552d687e74</checksum>
   ```

   Response Body

   When the operation completes successfully, the response returns an empty message body and a “204 No Content” status in the headers.

5. Use a GET request to view the updated zone “zone2” effective zoning configuration.

   URI

```
GET <base_URI>/rest/running/brocade-zone/effective-configuration/enabled-zone/zone-name/zone2
```

Request Body

No request body is required.

Response Body

The response contains a message body similar to the following, and has a “200 OK” status in the headers. The zone named “zone2” has been updated in the effective zoning configuration.

```
<?xml version="1.0"?>
<Response>
    <effective-configuration>
        <enabled-zone>
            <zone-name>zone2</zone-name>
            <zone-type>0</zone-type>
            <member-entry>
                <entry-name>10:00:00:00:00:00:00:04</entry-name>
                <entry-name>10:00:00:00:00:00:00:05</entry-name>
                <entry-name>10:00:00:00:00:00:00:06</entry-name>
                <entry-name>10:00:00:00:00:00:00:07</entry-name>
                <entry-name>10:00:00:00:00:00:00:08</entry-name>
                <entry-name>10:00:00:00:00:00:00:09</entry-name>
            </member-entry>
        </enabled-zone>
    </effective-configuration>
</Response>
```

Aborting a Zone Transaction

This example aborts a zone transaction before you commit a zone transaction.

This example assumes you have followed steps 1 through 4 of the 'Creating a zone using REST' or steps 1 through 3 of the 'Modifying a zone' use cases above.

1. Use a PATCH request to abort pending zone changes.

   URI

   ```
   PATCH <base_URI>/rest/running/brocade-zone/effective-configuration/cfg-action-v2/transaction-abort
   ```

   Request Body

   No request body is required.

   Response Body

   When the operation completes successfully, the response returns an empty message body and a “204 No Content” status in the headers.

2. Use a GET request to verify the pending zone changes are aborted.

   URI

   ```
   GET <base_URI>/rest/running/brocade-zone/effective-configuration/transaction-token
   ```

   Request Body

   No request body is required.

Response Body

The response contains a message body similar to the following, and has a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <effective-configuration>
        <transaction-token>0</transaction-token>
    </effective-configuration>
</Response>
```

Deleting a Zone

In this example, we will delete an existing zone, "zone4", from the defined-configuration.

1. Use a GET request to retrieve the current zoneDB checksum value.

   URI

   ```
   GET <base_URI>/rest/running/brocade-zone/effective-configuration/checksum
   ```

   Request Body

   No request body is required.

   Response Body

   The response contains a message body similar to the following, and has a “200 OK” status in the first line of the headers.

   ```
   <?xml version="1.0"?>
   <Response>
       <effective-configuration>
           <checksum>5c37a7bfe10f77de3253080b174a3239</checksum>   <== record this value
       </effective-configuration>
   </Response>
   ```

2. Use a GET request to verify the current zoning defined-configuration.

   URI

   ```
   GET <base_URI>/rest/running/brocade-zone/defined-configuration/zone
   ```

   Request Body

   No request body is required.

   Response Body

   The response contains a message body similar to the following, and has a “200 OK” status in the first line of the headers.

   ```
   <?xml version="1.0"?>
   <Response>
       <zone>
           <zone-name>zone1</zone-name>
           <zone-type>0</zone-type>
           <member-entry>
               <entry-name>10:00:00:00:00:00:00:01</entry-name>
               <entry-name>10:00:00:00:00:00:00:02</entry-name>
               <entry-name>10:00:00:00:00:00:00:03</entry-name>
   ```

```
            </member-entry>
        </zone>
        <zone>
            <zone-name>zone2</zone-name>
            <zone-type>0</zone-type>
            <member-entry>
                <entry-name>10:00:00:00:00:00:00:04</entry-name>
                <entry-name>10:00:00:00:00:00:00:05</entry-name>
                <entry-name>10:00:00:00:00:00:00:06</entry-name>
            </member-entry>
        </zone>
        <zone>
            <zone-name>zone3</zone-name>
            <zone-type>0</zone-type>
            <member-entry>
                <entry-name>10:00:00:00:00:00:00:07</entry-name>
                <entry-name>10:00:00:00:00:00:00:08</entry-name>
                <entry-name>10:00:00:00:00:00:00:09</entry-name>
            </member-entry>
        </zone>
        <zone>
            <zone-name>zone4</zone-name>
            <zone-type>0</zone-type>
            <member-entry>
                <entry-name>10:00:00:00:00:00:00:10</entry-name>
                <entry-name>10:00:00:00:00:00:00:11</entry-name>
                <entry-name>10:00:00:00:00:00:00:12</entry-name>
            </member-entry>
        </zone>
    </Response>
```

3. Use a DELETE request to delete the existing zone “zone4”.

   URI

   ```
   DELETE <base_URI>/rest/running/brocade-zone/defined-configuration/zone/zone-name/zone4
   ```

   Request Body

   No request body is required.

   Response Body

   When the request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

4. Use a PATCH request to save the zone configuration using the zoneDB checksum from step 1.

   URI

   ```
   PATCH <base_URI>/rest/running/brocade-zone/effective-configuration/cfg-action-v2/save
   ```

   Request Body

   ```
   <checksum>5c37a7bfe10f77de3253080b174a3239</checksum>
   ```

   Response Body

When the request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

5. Use a GET request to view the new defined zoning configuration.

URI

```
GET <base_URI>/rest/running/brocade-zone/defined-configuration/zone
```

Request Body

No request body is required.

Response Body

The response contains a message body similar to the following, and has a “200 OK” status in the first line of the headers. The zone named “zone4” has been removed from the zoning configuration named “cfg1”.

```
<?xml version="1.0"?>
<Response>
    <zone>
        <zone-name>zone1</zone-name>
        <zone-type>0</zone-type>
        <member-entry>
            <entry-name>10:00:00:00:00:00:00:01</entry-name>
            <entry-name>10:00:00:00:00:00:00:02</entry-name>
            <entry-name>10:00:00:00:00:00:00:03</entry-name>
        </member-entry>
    </zone>
    <zone>
        <zone-name>zone2</zone-name>
        <zone-type>0</zone-type>
        <member-entry>
            <entry-name>10:00:00:00:00:00:00:04</entry-name>
            <entry-name>10:00:00:00:00:00:00:05</entry-name>
            <entry-name>10:00:00:00:00:00:00:06</entry-name>
        </member-entry>
    </zone>
    <zone>
        <zone-name>zone3</zone-name>
        <zone-type>0</zone-type>
        <member-entry>
            <entry-name>10:00:00:00:00:00:00:07</entry-name>
            <entry-name>10:00:00:00:00:00:00:08</entry-name>
            <entry-name>10:00:00:00:00:00:00:09</entry-name>
        </member-entry>
    </zone>
</Response>
```

Concurrent Zoning Transactions on a Local Switch

In this example, assume that there are two FOS API REST clients (R1a and R1b) logged in to the same switch.

FOS API REST client 1a (R1a) opens a zone transaction normally. If FOS API REST client 1b (R1b) also attempts to open a zone transaction, the zone operation attempt from R1b will fail with the following error:

```
<?xml version="1.0"?>
<errors xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
```

```
    <error>
        <error-type>protocol</error-type>
        <error-tag>Operation-failed</error-tag>
        <error-app-tag>Error</error-app-tag>
        <error-path>/zone/zone-name/rest1b_zone1/</error-path>
        <error-message>There is an outstanding REST transaction, and you are not the owner of that
 transaction. (4 mins 48 secs left)
</error-message>
        <error-info>
            <error-code>-3</error-code>
            <error-module>zone</error-module>
        </error-info>
    </error>
</errors>
```

Because R1a has an open zone transaction and is still logged in, the R1a client transaction will not be disturbed as long as that client is logged in and the session has not reached the configured timeout limit.

Commitment of Simultaneous Multiswitch Zone Transactions

For this example, assume that there are two switches in a fabric (Switch 1 and Switch 2).

1. A FOS API REST client (Rest_sw1) starts a zone transaction on Switch 1.
2. A different FOS API REST client (Rest_sw2) starts a zone transaction on Switch 2.
3. Both clients the attempt to commit their changes at the same time.

In this case, whichever RESTCONF transaction is processed first would win and go through. In the event of an exact tie, both commits would be rejected and failed. The clients would have to retry, and whoever got their retry attempt in first would win. The client who loses would have its pending transaction aborted as a result of the winning REST client's commit operation.

Timeout of the REST Zone Transaction Timer

For this example, assume that FOS REST API client 1 is logged in to Switch 1.

1. A FOS REST API client on Switch 1 (Rest_sw1) starts a zone transaction.
2. The Rest_sw1 user leaves its zone transaction open with no activity occurring for more than the configured lock timeout value.
3. At this point, a CLI user on Switch 1 (CLI_sw1) attempts a zone transaction. Because the Rest_sw1 timer has expired, the CLI_sw1 transaction is allowed to be opened, and as a result, Rest_sw1's transaction is cancelled.
4. If Rest_sw1 later attempts to continue its zone transactions by issuing another zone operation, it will see the following error:

```
 <errors xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
    <error>
        <error-type>protocol</error-type>
        <error-tag>Operation-failed</error-tag>
        <error-app-tag>Error</error-app-tag>
        <error-path>/zone/zone-name/rest_sw1_zone2/</error-path>
        <error-message>Warning: Cannot complete operation due to the current zoning transaction being
 aborted (Reason: User Command).  Retry prior operation(s) to recover.</error-message>
        <error-info>
            <error-code>-16</error-code>
            <error-module>zone</error-module>
        </error-info>
    </error>
```

```
</errors>
```

REST Running Config

This section details the available REST running configs in the FOS REST API. This section also provides information about the parameters that are associated with the REST running configs.

brocade-access-gateway/device-list

Description

A list of devices logged in via the Access Gateway and their F_Port,

N_Port, and FCID mapping.

Parameters

Table 10: brocade-access-gateway/device-list

Name	Type	Description
wwn	fibrechannel:wwn-type	The port world wide name (WWN) of the connected device.
fcid	fibrechannel:fcid-hex-string-type	The Fibre Channel ID (FCID) of the connected device.
f-port	fibrechannel:slot-port-name-type	The F_Port to which the device is connected.
n-port	fibrechannel:slot-port-name-type	The N_Port through which the device logged in.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <f-port-list>
        <f-port>0/0</f-port>
        <online-status>0</online-status>
        <operational-status>offline</operational-status>
        <f-port-info/>
```

```
    </f-port-list>
...
    <f-port-list>
        <f-port>0/15</f-port>
        <online-status>0</online-status>
        <operational-status>offline</operational-status>
        <f-port-info/>
    </f-port-list>
</Response>
```

brocade-access-gateway/f-port-list

Description

List of all F_Ports present on the Access Gateway.

Parameters

Table 11: brocade-access-gateway/f-port-list

Name	Type	Description
f-port	fibrechannel:slot-port-name-type	The F_Port for which information is being fetched.
online-status	uint8	Whether the F_Port is online or offline. 0: Offline. 1: Online. Deprecated: For read operations, this leaf will be returned in responses with valid data comparable to the 'operational-status' leaf.
operational-status	access-gateway-type:operational-status-type	Whether the F_Port is online or offline.
f-port-info	Container	The F_Port login information.
- n-port	fibrechannel:slot-port-name-type	The N_Port to which this F_Port is mapped.
- login-exceeded	uint8	The login exceeded state of the F_Port. 0 : Login limit not exceeded. 1 : Login limit exceeded. Deprecated: For read operations, this leaf will be returned in responses with valid data comparable to the 'login-exceeded-v2' leaf.
- login-exceeded-v2	boolean	The login exceeded state of the F_Port. false : Login limit not exceeded. true : Login limit exceeded.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "f-port-list": [
            {
                "f-port": "0/0",
                "online-status": 0,
                "operational-status": "offline",
                "f-port-info": {}
            },
            {
                "f-port": "0/1",
                "online-status": 0,
                "operational-status": "offline",
                "f-port-info": {}
            },
            {
                "f-port": "0/2",
                "online-status": 0,
                "operational-status": "offline",
                "f-port-info": {}
            },

...
            {
                "f-port": "0/62",
                "online-status": 0,
                "operational-status": "offline",
                "f-port-info": {}
            },
            {
                "f-port": "0/63",
                "online-status": 0,
                "operational-status": "offline",
                "f-port-info": {}
            }
        ]
    }
}
```

brocade-access-gateway/n-port-settings

Description

The N_Port-related configuration parameters.

Parameters

Table 12: brocade-access-gateway/n-port-settings

Name	Type	Description
reliability-counter	yang:zero-based-counter64	The reliability counter configuration for N_Ports. 0 : Disabled. 10-100 : Enabled.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <n-port-settings>
        <reliability-counter>25</reliability-counter>
    </n-port-settings>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-access-gateway/policy

Description

The Access Gateway policy configuration. To enable or disable any of

the AG policies, the switch must be in a disabled state.

User cannot modify both the Port Group and Auto Policy in a single

request. Updates to the Port Group and Auto Policy must be performed

via separate requests.

Parameters

Table 13: brocade-access-gateway/policy

Name	Type	Description
port-group-policy-enabled	uint8	Enables or disables the port group policy. 0 : Disabled. 1 : Enabled. Deprecated: For read operations, this leaf will be returned in responses with valid data comparable to the 'port-group-policy -enabled-v2' leaf.
port-group-policy-enabled-v2	boolean	Enables or disables the port group policy. false : Disabled. true : Enabled.
auto-policy-enabled	uint8	Enables or disables the auto policy. 0 : Disabled. 1 : Enabled. Disabling auto policy will enable port-group-policy as default. Deprecated: For read operations, this leaf will be returned in responses with valid data comparable to the 'auto-policy-enabled-v2' leaf.
auto-policy-enabled-v2	boolean	Enables or disables the auto policy. false : Disabled. true : Enabled. Disabling auto policy will enable port-group-policy as default.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "policy": {
            "port-group-policy-enabled": 0,
            "port-group-policy-enabled-v2": false,
            "auto-policy-enabled": 1,
            "auto-policy-enabled-v2": true
        }
```

```
        }
    }
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-access-gateway/port-group

Description

The port group configuration.

The port group defines a set of N_Ports to be included in the Port Grouping policy. The factory default port group is '0', which includes all N_Ports. The default port group cannot be removed or renamed.

Parameters

Table 14: brocade-access-gateway/port-group

Name	Type	Description
port-group-id	uint8	The port group ID. The maximum number of port groups that can be created is equal to the total number of ports available on the given platform. For instance, on a 64-port platform, there can be 64 port groups with the port-group ID ranging from 0 to 63.
port-group-name	string	The port group name. The name must be alphanumeric. The default port group name uses the format pg<port-group-id>.
port-group-n-ports	Container	List of N_Ports that are in the port group.
- n-port	fibrechannel:slot-port-name-type	List of N_ports that are in the port group.
port-group-f-ports	Container	List of F_Ports that are in the port group.
- f-port	fibrechannel:slot-port-name-type	List of F_Ports that are in the port group. To update the configuration, you must enable load-balancing mode for the port group.
port-group-mode	Container	Different modes of operation of the port-group
- load-balancing-mode-enabled	uint8	Enables or disables load-balancing mode for the specified port group. 0 : Load-balancing mode disabled. 1 : Load-balancing mode enabled. Deprecated: For read operations, this leaf will be returned in responses with valid data comparable to the 'load-balancing-mode-enabled-v2' leaf.
- load-balancing-mode-enabled-v2	boolean	Enables or disables load-balancing mode for the specified port group. false : Load-balancing mode disabled. true : Load-balancing mode enabled.

Name	Type	Description
- multiple-fabric-name-monitoring-mode-enabled	uint8	Enables or disables multiple fabric name monitor mode for the specified port group. 0 : Multiple fabric name monitor mode disabled. 1 : Multiple fabric name monitor mode enabled. Deprecated: For read operations, this leaf will be returned in responses with valid data comparable to the 'multiple-fabric-name -monitoring-mode-enabled-v2' leaf.
- multiple-fabric-name-monitoring-mode-enabled-v2	boolean	Enables or disables multiple fabric name monitor mode for the specified port group. false : Multiple fabric name monitor mode disabled. true : Multiple fabric name monitor mode enabled.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <port-group>
  <port-group-id>0</port-group-id>
  <port-group-name>pg0</port-group-name>
  <port-group-n-ports>
   <n-port>0/16</n-port>
   <n-port>0/17</n-port>
   <n-port>0/18</n-port>
   <n-port>0/19</n-port>
   <n-port>0/20</n-port>
   <n-port>0/21</n-port>
   <n-port>0/22</n-port>
   <n-port>0/23</n-port>
  </port-group-n-ports>
  <port-group-f-ports>
   <f-port>0/0</f-port>
   <f-port>0/1</f-port>
   <f-port>0/2</f-port>
   <f-port>0/3</f-port>
   <f-port>0/4</f-port>
   <f-port>0/5</f-port>
   <f-port>0/6</f-port>
   <f-port>0/7</f-port>
```

```
    <f-port>0/8</f-port>
    <f-port>0/9</f-port>
    <f-port>0/10</f-port>
    <f-port>0/11</f-port>
    <f-port>0/12</f-port>
    <f-port>0/13</f-port>
    <f-port>0/14</f-port>
    <f-port>0/15</f-port>
  </port-group-f-ports>
  <port-group-mode>
    <load-balancing-mode-enabled>0</load-balancing-mode-enabled>
    <multiple-fabric-name-monitoring-mode-enabled>0</multiple-fabric-name-monitoring-modeenabled>
   </port-group-mode>
  </port-group>
 </Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-application-server/application-server-device

Description

A list of application identifier allocations.

This list supports the following query parameter combinations:

n-port-id

n-port-id + application-id

entity-id-ascii

n-port-id + entity-id-ascii

Parameters

Table 15: brocade-application-server/application-server-device

Name	Type	Description
n-port-id	fibrechannel:fcid-hex-string-type	The N_Port ID (Fibre Channel address) of the device. This leaf can be queried.
application-id	uint32	The application identifier that has been allocated to the entity identifier for a given N_Port ID. This leaf can be queried.
entity-id-hex	fibrechannel:brocade-hex-string-type	The entity identifier as hexadecimal number.
entity-id-ascii	string	The entity identifier as an ASCII string. If the registered value is not within a printable ASCII range, then the response is empty (NULL). This leaf can be queried.
entity-name	string	An alias name for the entity. The registration of this attribute is optional.
host-identifier	string	The common host name associated with the entity. The registration of this attribute is optional.
symbolic-data	string	The symbolic information associated with the entity. The registration of this attribute is optional.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-chassis/chassis

Description

The complete details of the chassis.

Parameters

Table 16: brocade-chassis/chassis

Name	Type	Description
chassis-user-friendly-name	fibrechannel:chassis-user-friendly-name-type	An ASCII name assigned to the switch chassis by the administrator. A chassis name must begin with a letter and can consist of letters, numbers, underscores (_) or hyphens (-) and is not case-sensitive. In FICON mode, the name is limited to a maximum of 24 characters plus hyphens (-) and underscores (_).
chassis-wwn	fibrechannel:wwn-type	The WWN of the chassis.
license-id	fibrechannel:wwn-type	The WWN which is used for generating the license.
manufacturer	fru:manufacturer-type	The manufacturer of the chassis.
registered-organization-name	string	Displays the registered organization name configured in the switch.
organization-registration-date	string	Displays the date of completion of the registration. Registration date is of mmm 20YY format (Example: may 2015).
part-number	fru:part-number-type	The part number for the physical element assigned by the manufacturer. Deprecated: Please refer 'brocade-fru.yang' 'list wwn' 'part-number' leaf instead.
serial-number	fru:serial-number-type	A printable ASCII string that specifies the factory serial number of the chassis.
entitlement-serial-number	fru:serial-number-type	A serial number which is used for entitlement support.
max-blades-supported	uint16	The maximum number of blades that can fit in the physical chassis. This number includes switch, control processor, application, and core routing blades.
vendor-serial-number	string	The serial number of the chassis assigned by the vendor. Deprecated: Please refer 'brocade-fru.yang' 'list wwn' 'vendor-serial-number' leaf instead.
vendor-part-number	string	The part number of the chassis assigned by the vendor. Deprecated: Please refer 'brocade-fru.yang' 'list wwn' 'vendor-part-number' leaf instead.
vendor-revision-number	string	The revision number of the chassis assigned by the vendor. Deprecated: Please refer 'brocade-fru.yang' 'list wwn' 'vendor-revision-number' leaf instead.
product-name	fibrechannel:product-name-type	The product name of the chassis.
vf-enabled	boolean	Whether Virtual Fabrics is enabled on the chassis. Enabling Virtual Fabrics is a disruptive operation that requires a reboot to take effect. true : Virtual Fabrics is enabled on the chassis. false : Virtual Fabrics is not enabled on the chassis.
vf-supported	boolean	Whether Virtual Fabrics is supported on the chassis. true : Virtual Fabrics is supported on the chassis. false : Virtual Fabrics is not supported on the chassis.

Name	Type	Description
fcr-enabled	boolean	Indicates whether FCR is enabled or not. true : enabled false : disabled
fcr-supported	boolean	Indicates whether FCR is supported on the chassis or not. true : FCR is supported false : FCR is not supported
date	string	The current date of the switch.
chassis-enabled	boolean	Indicates whether chassis is in enabled or disabled state. true : chassis is enabled. false : chassis is disabled.
message-of-the-day	string	Specifies the text string of alphanumeric characters to be displayed as the banner on the chassis before login. The message displays either before or after the login prompt. Fabric OS does not control when the MOTD displays. The maximum length is 1022 characters.
shell-timeout	uint32	Bash timeout value. Setting this to 0 disables bash timeout
session-timeout	uint32	SSH Session timeout value. Setting this to 0 disables ssh session timeout.
usb-device-enabled	boolean	Indicates whether USB is in enabled or disabled state. true : USB device is enabled. false : USB device is disabled.
usb-available-space	uint32	Available space on the USB connected device.
tcp-timeout-level	uint8	Display or configure the timeout level for half open TCP IPv4 and IPv6 connections in all the management ethernet interfaces in switch. Approximate timeout value for each level is provided as below: Timeout level | Timeout in seconds 1 | 3 2 | 7 3 | 15 4 | 31 5 | 63
backplane-revision	fibrechannel:brocade-hex-string-type	The chassis backplane revision number.
system-uptime	yang:timeticks	The period of time elapsed since last reboot of the specified switch. This value represents an absolute time interval for which the switch has been up, not the time measured between the last reboot and midnight, January 1, 1970 UTC, and the most recent reboot of the switch.
user-sessions	uint32	The number of active user sessions on the chassis.
one-minute-load-average	decimal64	The average load on the CPU for the last one minute.
five-minutes-load-average	decimal64	The average load on the CPU for the last five minutes.
fifteen-minutes-load-average	decimal64	The average load on the CPU for the last fifteen minutes.

Name	Type	Description
total-memory	yang:gauge64	Total usable RAM. Does not include swap space.
used-memory	yang:gauge64	Memory currently in use. Excludes memory used by Kernel buffers, page cache and slabs.
free-memory	yang:gauge64	Sum of unused Low memory and High memory.
shared-memory	yang:gauge64	Amount of memory consumed in tmpfs type file systems.
buffer-cache-memory	yang:gauge64	Sum of memory used by Kernel buffers, page cache, slabs.
available-memory	yang:gauge64	Estimate memory available to start new applications without swapping.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "chassis": {
            "chassis-user-friendly-name": "X6-4_Venator",
            "license-id": "10:00:c4:f5:7c:4a:ac:6c",
            "chassis-wwn": "10:00:c4:f5:7c:4a:ac:eb",
            "serial-number": "EZB2520M004",
            "entitlement-serial-number": "EZB2520M004",
            "manufacturer": "Brocade Communications Systems LLC",
            "registered-organization-name": "sqa",
            "organization-registration-date": "sep 2017",
            "part-number": "60-1003194-02",
            "vf-enabled": false,
            "vf-supported": true,
            "fcr-enabled": true,
            "fcr-supported": true,
            "max-blades-supported": 8,
            "vendor-revision-number": "25",
            "vendor-part-number": "SLKWRM0000X4S",
            "vendor-serial-number": "infodot-212618",
            "product-name": "x6-4",
            "chassis-enabled": true,
            "shell-timeout": 0,
            "session-timeout": 0,
            "message-of-the-day": "123",
            "date": "03/01/2023-13:45:28",
            "usb-device-enabled": false,
```

```
            "usb-available-space": 0,
            "tcp-timeout-level": 5,
            "system-uptime": 22428,
            "user-sessions": 2,
            "one-minute-load-average": 0.15,
            "five-minutes-load-average": 0.23,
            "fifteen-minutes-load-average": 0.25,
            "total-memory": 15989196,
            "used-memory": 962660,
            "free-memory": 14481168,
            "shared-memory": 428624,
            "buffer-cache-memory": 545368,
            "available-memory": 13708520,
            "backplane-revision": "0x10"
        }
    }
}
```


PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-chassis/credit-recovery

Description

The credit loss detection and recovery for backend ports

in the Director chassis.

Parameters

Table 17: brocade-chassis/credit-recovery

Name	Type	Description
mode	enumeration	Internal port credit recovery configuration modes.
link-reset-threshold	uint16	When the number of Link Reset per hour reaches the configured threshold and if the mode is Link Reset Threshold, the action to fault the blade based on fault option will be executed.
fault-option	enumeration	Fault option for Backend Link failure.

Name	Type	Description
backend-credit-loss-enabled	boolean	Enable or Disable link reset for credit loss alone. True enables link reset when loss of credit happens and mode needs to be set to link-reset-only or link-reset-threshold before this operation. False disables the link reset recovery when loss of credit. When the value of the attribute backend-credit-loss-enabled is false, the leaf attribute mode will not be overwritten but will be unused internally and hence will be displayed as off when GET instance on leaf attribute mode happens. When the value of the attribute backend-credit-loss-enabled is true, the current configured value will be shown for leaf attribute mode as part of the GET instance. When attribute “backend-credit-loss-enabled” is specified in the input, it will take the precedence over attribute “mode”.
backend-loss-of-sync-enabled	boolean	Enable or Disable link reset for loss of sync alone. True enables link reset for loss of sync and mode needs to be set to link-reset-only or link-reset-threshold before this operation. False disables the link reset recovery for loss of sync.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <credit-recovery>
        <mode>off</mode>
        <link-reset-threshold>10</link-reset-threshold>
        <fault-option>edge-core-blade</fault-option>
        <backend-credit-loss-enabled>false</backend-credit-loss-enabled>
        <backend-loss-of-sync-enabled>false</backend-loss-of-sync-enabled>
    </credit-recovery>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-chassis/ha-status

Description

The control processor (CP) status, which includes the following information:

- Local CP state (slot number and CP ID) and warm or cold
- Remote CP state (slot number and CP ID)
- High Availability (enabled or disabled)
- Heartbeat (up or down)
- Health of the standby CP
- HA synchronization status

Parameters

Table 18: brocade-chassis/ha-status

Name	Type	Description
active-cp	fibrechannel:cp-name-type	The ID of the active CP.
standby-cp	fibrechannel:cp-name-type	The ID of the standby CP.
active-slot	uint16	The slot number of the active CP.
standby-slot	uint16	The slot number of the standby CP.
recovery-type	string	The recovery status of the switch.
standby-health	string	The health status of the standby CP. healthy : The standby CP is running, and the background health diagnostic has not detected any errors. faulted : The standby CP is running, but the background health diagnostic has discovered a problem with the blade. Check the logs to determine the appropriate action. Failover is disabled until the standby CP is repaired. unknown : The standby CP health state is unknown because of one of the following reasons: the standby CP does not exist, the heartbeat is down, or the Health Monitor has detected a configuration file error. non-redundant : There is no standby CP. not-available : The standby CP health status is not available.
ha-enabled	boolean	Whether the Availability (HA) is enabled or disabled. true : HA is enabled. false : HA is disabled.

Name	Type	Description
heartbeat-up	boolean	Whether the heartbeat to standby CP is up or down. true : The heartbeat to standby CP is up. false : The heartbeat to standby CP is down.
ha-synchronized	boolean	Whether HA is in a synchronized state. When HA is in a synchronized state and a failover becomes necessary, it is nondisruptive. true : HA is synchronized. false : HA is not synchronized.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <ha-status>
  <active-cp>cp0</active-cp>
  <standby-cp>cp1</standby-cp>
  <active-slot>1</active-slot>
  <standby-slot>2</standby-slot>
  <recovery-type>warm</recovery-type>
  <standby-health>healthy</standby-health>
  <ha-enabled>true</ha-enabled>
  <heartbeat-up>true</heartbeat-up>
  <ha-synchronized>true</ha-synchronized>
 </ha-status>
</Response>
```

brocade-chassis/management-ethernet-interface

Description

A list of management ethernet interface configurations

for fixed-port and chassis switches.

Parameters

Table 19: brocade-chassis/management-ethernet-interface

Name	Type	Description
cp-name	fibrechannel:cp-name-type	Identification of host CP in fixed-port and chassis switch.
interface-name	enumeration	Name of the interface in CP for fixed-port and chassis switch. Valid value for interface name in fixed-port switch is eth0. Valid values for interface name in chassis are eth0, eth3 and bond0.
auto-negotiate	boolean	Enabled or disabled status of auto-negotiation feature for ethernet link. true - Auto-negotiation is/will be enabled. false - Auto-negotiation is/will be disabled. This is not a configurable parameter on certain platforms.
speed	enumeration	Speed of the ethernet link in Mbps. When auto-negotiate is enabled, the response includes this leaf only when the ethernet link connection is established. Note: When auto-negotiate is 'false', speed must be specified, when auto-negotiate is 'true' speed is optional. This is not a configurable parameter on certain platforms.
duplex	enumeration	Capability of the ethernet link. When auto-negotiate is enabled, the response includes this leaf only when the ethernet link connection is established.
connection-established-status	boolean	Ethernet Link connection if established is true else false.
ethernet-mac-address	yang:mac-address	MAC Address of ethernet link.
active-interface	union {fibrechannel:physical-interface-type;string;}	Current active interface in bond0. Applicable only for chassis switches.
inet-address	inet:ipv4-address	IPv4 address for the interface. The response includes this leaf only for eth0 interface in fixed-port switches and bond0 interface in chassis switches.
broadcast-address	inet:ipv4-address	Broadcast address for the interface. The response includes this leaf only for eth0 interface in fixed-port switches and bond0 interface in chassis switches.
subnet-mask	inet:ipv4-address	Network mask for the interface. The response includes this leaf only for eth0 interface in fixed-port switches and bond0 interface in chassis switches.
link-inet6-address	union {inet:ipv6-address;inet:ipv6-prefix;}	Link local address of the interface. The response includes this leaf only for eth0 interface in fixed-port switches and bond0 interface in chassis switches.
mtu	uint32	Maximum transmission unit in bytes for the ethernet interface. Normally it is 1500.
metric	yang:gauge32	Cost of using the interface.
rx-packets	yang:zero-based-counter64	Total number of packets received.
rx-errors	yang:zero-based-counter64	Total number of packets received with error.
rx-dropped	yang:zero-based-counter64	Total number of dropped packets during reception.

Name	Type	Description
rx-overruns	yang:zero-based-counter64	Total number of received packets that faced fifo overruns.
rx-frame	yang:zero-based-counter64	Total number of misaligned packets received.
rx-bytes	yang:zero-based-counter64	Total number of bytes received from interface.
tx-packets	yang:zero-based-counter64	Total number of packets transmitted.
tx-errors	yang:zero-based-counter64	Total number of packets transmitted with error.
tx-dropped	yang:zero-based-counter64	Total number of dropped packets during transmission.
tx-overruns	yang:zero-based-counter64	Total number of transmitted packets that faced fifo overruns.
tx-carrier	yang:zero-based-counter64	Total number of packets that experienced loss of carriers.
tx-collisions	yang:zero-based-counter64	Total number of transmitted packets that experienced ethernet collisions.
tx-queue-length	uint32	Length of the transmission queue in unit of packets
tx-bytes	yang:zero-based-counter64	Total number of bytes transmitted over interface.
start-memory-address	fibrechannel:brocade-hex-string-type	Start address of memory segment that onboard memory segment of NIC is mapped into. It is optional and may be absent in certain platforms.
end-memory-address	fibrechannel:brocade-hex-string-type	End address of memory segment that onboard memory segment of NIC is mapped into. It is optional and may be absent in certain platforms.
lldp-enabled	boolean	LLDP state of the port. Default state is true. Possible values are: true - LLDP is enabled on the port false - LLDP is disabled on the port
standby-interfaces	Container	List of one or two currently standby interfaces available in bond0.
- interface	fibrechannel:physical-interface-type	Currently standby interface in bond0. Applicable only for chassis switches.
global-inet6-addresses	Container	List of IPv6 addresses with global scope assigned to the ethernet interface of the switch. The response includes this container only when global IPv6 addresses are configured for eth0 interface in fixed-port switches and bond0 interface in chassis switches.
- global-inet6-address	union {inet:ipv6-address;inet:ipv6-prefix;}	IPv6 address of global scope assigned to the ethernet interface.
ethernet-status-flags	Container	List of configuration flags for the ethernet interface.
- flag	enumeration	Flags indicating the configuration status of ethernet interface.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <management-ethernet-interface>
        <cp-name>cp0</cp-name>
        <interface-name>eth0</interface-name>
        <auto-negotiate>true</auto-negotiate>
        <speed>1000BASE-T</speed>
        <duplex>full</duplex>
        <connection-established-status>true</connection-established-status>
        <ethernet-mac-address>C1:F1:1C:11:A1:D1</ethernet-mac-address>
        <ethernet-status-flags>
            <flag>up</flag>
            <flag>broadcast</flag>
            <flag>running</flag>
            <flag>slave</flag>
            <flag>multicast</flag>
        </ethernet-status-flags>
        <mtu>1500</mtu>
        <metric>1</metric>
        <rx-packets>5063786</rx-packets>
        <rx-errors>0</rx-errors>
        <rx-dropped>7</rx-dropped>
        <rx-overruns>0</rx-overruns>
        <rx-frame>0</rx-frame>
        <rx-bytes>804103934</rx-bytes>
        <tx-packets>1263525</tx-packets>
        <tx-errors>0</tx-errors>
        <tx-dropped>0</tx-dropped>
        <tx-overruns>0</tx-overruns>
        <tx-carrier>0</tx-carrier>
        <tx-collisions>0</tx-collisions>
        <tx-queue-length>1000</tx-queue-length>
        <tx-bytes>473658802</tx-bytes>
        <start-memory-address>0xfe5e6000</start-memory-address>
        <end-memory-address>0xfe5e6fff</end-memory-address>
        <lldp-enabled>true</lldp-enabled>
    </management-ethernet-interface>
...
    <management-ethernet-interface>
        <cp-name>cp1</cp-name>
        <interface-name>bond0</interface-name>
        <auto-negotiate>true</auto-negotiate>
        <speed>1000BASE-T</speed>
        <duplex>full</duplex>
        <connection-established-status>true</connection-established-status>
        <ethernet-mac-address>C2:F2:2C:22:AC:21</ethernet-mac-address>
        <active-interface>eth0</active-interface>
```

```
        <standby-interfaces>
            <interface>eth3</interface>
        </standby-interfaces>
        <inet-address>10.10.10.20</inet-address>
        <broadcast-address>10.10.10.255</broadcast-address>
        <subnet-mask>255.255.255.0</subnet-mask>
        <global-inet6-addresses>
            <global-inet6-address>2120:100:1:fa01::f031/128</global-inet6-address>
            <global-inet6-address>2120:100:1:fa01:c6f1:7cff:fe11:ac91/64</global-inet6-address>
        </global-inet6-addresses>
        <link-inet6-address>fe20::c6f2:7cff:fe22:ac21/24</link-inet6-address>
        <ethernet-status-flags>
            <flag>up</flag>
            <flag>broadcast</flag>
            <flag>running</flag>
            <flag>master</flag>
            <flag>multicast</flag>
        </ethernet-status-flags>
        <mtu>1500</mtu>
        <metric>1</metric>
        <rx-packets>3819433</rx-packets>
        <rx-errors>0</rx-errors>
        <rx-dropped>6</rx-dropped>
        <rx-overruns>0</rx-overruns>
        <rx-frame>0</rx-frame>
        <rx-bytes>633442022</rx-bytes>
        <tx-packets>73962</tx-packets>
        <tx-errors>0</tx-errors>
        <tx-dropped>0</tx-dropped>
        <tx-overruns>0</tx-overruns>
        <tx-carrier>0</tx-carrier>
        <tx-collisions>0</tx-collisions>
        <tx-queue-length>0</tx-queue-length>
        <tx-bytes>6128167</tx-bytes>
    </management-ethernet-interface>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-chassis/management-interface-configuration

Description

Configures the maximum number of REST sessions, throttling configurations for incoming REST requests, enable or disable keep-alive connection.

Parameters

Table 20: brocade-chassis/management-interface-configuration

Name	Type	Description
rest-enabled	boolean	This parameter is used to disable the REST interface. If REST is already disabled, it can only be enabled through CLI.
https-protocol-enabled	boolean	This parameter is used to enable HTTP protocol if certificates are not installed and HTTPS is not effective. This configuration has no effect if HTTPS is effective as shown by 'effective-protocol' below.
effective-protocol	enumeration	Protocol currently in effect for REST communication, HTTP or HTTPS.
max-rest-sessions	uint32	Maximum number of REST sessions that are allowed. The Default number of sessions is 3. New logins denied, if the session count has reached maximum limit.
https-keep-alive-enabled	boolean	This parameter is used to enable or disable keep-alive connection in HTTPS.
https-keep-alive-timeout	uint32	The keep-alive connection expires after an idle time of 15 seconds.
authentication-mode-enabled	boolean	This parameter is used to enable or disable the RBAC check for CAL/ HTML request. true - RBAC authentication enabled. false - RBAC authenticaiton disabled.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <management-interface-configuration>
        <rest-enabled>true</rest-enabled>
        <https-protocol-enabled>true</https-protocol-enabled>
        <effective-protocol>https</effective-protocol>
        <max-rest-sessions>10</max-rest-sessions>
        <https-keep-alive-enabled>false</https-keep-alive-enabled>
        <https-keep-alive-timeout>15</https-keep-alive-timeout>
    </management-interface-configuration>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-chassis/management-port-connection-statistics

Description

Details regarding the ethernet connections on management port.

This list supports the following query parameter combinations:

cp-name

protocol

local-port

remote-port

ip-family

operational-status

Parameters

Table 21: brocade-chassis/management-port-connection-statistics

Name	Type	Description
cp-name	fibrechannel:cp-name-type	Specifies the name of the CP to which the connection belongs to.
ip-family	inet:ip-version	Specifies the internet protocol version that connection belongs to.
operational-status	enumeration	Specifies if the connection is currently active or not.
local-ip-address	inet:ip-address	Specifies the local/source IPv4/IPv6 address of the network connection.
local-port	inet:port-number	Specifies the local/source port number of the network connection.
remote-ip-address	inet:ip-address	Specifies the remote/destination IPv4/IPv6 address of the network connection.
remote-port	inet:port-number	Specifies the remote/destination port number of the network connection.
start-time	yang:date-and-time	Timestamp when the socket connection was opened at the switch.
state-update-time	union {yang:date-and-time;fibrechannel:brocade-empty-string-type;}	Timestamp when the socket connection state changed.
protocol	string	Represents the transport connection protocol used.
connection-state	brocade-interface-types:connection-state-type	State of the connection.
process-name	string	Name of the process that opened the socket.
process-id	uint16	Process id of the program that is responsible for the connection.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?<
<Response<
    <management-port-connection-statistics<
        <cp-name<cp0</cp-name<
        <ip-family<ipv6</ip-family<
        <operational-status<active</operational-status<
        <local-ip-address<fe20::c2f2:2cff:fe22:a2d2</local-ip-address<
        <local-port<546</local-port<
        <remote-ip-address<::</remote-ip-address<
        <remote-port<0</remote-port<
        <start-time<2021-11-05T11:52:39-06:00</start-time<
        <state-update-time/<
        <protocol<udp</protocol<
        <connection-state<closed</connection-state<
        <process-name<dhclient</process-name<
        <process-id<30007</process-id<
    </management-port-connection-statistics<
...
    <management-port-connection-statistics<
        <cp-name<cp1</cp-name<
        <ip-family<ipv4</ip-family<
        <operational-status<inactive</operational-status<
        <local-ip-address<10.10.10.142</local-ip-address<
        <local-port<22</local-port<
        <remote-ip-address<10.10.10.14</remote-ip-address<
        <remote-port<59056</remote-port<
        <start-time<2021-11-05T14:25:50-06:00</start-time<
        <state-update-time<2021-11-05T14:26:50-06:00</state-update-time<
        <protocol<tcp</protocol<
        <connection-state<established</connection-state<
        <process-name<sshd</process-name<
        <process-id<12992</process-id<
    </management-port-connection-statistics<
</Response<
```

brocade-chassis/management-port-track-configuration

Description

Ethernet connections management port tracker configuration.

Parameters

Table 22: brocade-chassis/management-port-track-configuration

Name	Type	Description
tracking-enabled	boolean	Specifies if the management port tracking is enabled or disabled. true: enabled false: disabled
scan-interval	uint16	Specifies the interval at which the kernel is queried for the active network connection at the switch;

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <management-port-track-configuration>
        <tracking-enabled>true</tracking-enabled>
        <scan-interval>30</scan-interval>
    </management-port-track-configuration>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-chassis/sn-chassis

Description

SANnav Internal purpose,

New internal URI for REST bound application attribute,

The attribute provides the address of the server

which was bound to switch.

Parameters

Table 23: brocade-chassis/sn-chassis

Name	Type	Description
bound-application	inet:ip-address	It indicate SANnav binding server IP address which bounds to switch.
force	boolean	Indicates force bind of a server to switch, true = forcefully bind even switch is already bound to old server. false = bind only when no application was bound.

Produces

- application/xml
- application/json

brocade-chassis/version

Description

The top-level container for version details

of the switch.

Parameters

Table 24: brocade-chassis/version

Name	Type	Description
kernel	string	The version of switch kernel operating system.
fabric-os	string	The firmware version of the switch.
made-on	yang:date-and-time	The build date of firmware running in switch.
flash	yang:date-and-time	The build date of firmware stored in flash PROM(s).
boot-prom	string	The version of the firmware stored in the boot PROM.

Produces

- application/xml
- application/json

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
{
    "Response": {
        "version": {
            "kernel": "5.4.66_rt38",
            "fabric-os": "v9.2.0_bld57",
            "made-on": "2023-02-28T11:23:23-07:00",
            "flash": "2023-03-01T07:24:17-07:00",
            "boot-prom": "4.0.17-sb"
        }
    }
}
```


PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-extension-ip-route/extension-ip-route

Description

The static IP route on the IP interface defined on an

extension blade or system.

Parameters

Table 25: brocade-extension-ip-route/extension-ip-route

Name	Type	Description
name	brocade-interface-types:ip-extension-interface-type	The name of the interface.
dp-id	extension:dp-id-type	The Extension Data Path Processor ID associated with the IP Route. Depending on the platform it either has a single DP or dual DP. A single DP supports DP0 only. A dual DP supports both DP0 and DP1. 0 : DP0 1 : DP1
ip-address	union {inet:ip-address;inet:ip-prefix;}	The destination IPv4/IPv6 address.
ip-prefix-length	brocade-interface-types:ip-prefix-length	The prefix length operator for the destination IP address. Once set, the prefix length cannot be changed.
ip-gateway	brocade-interface-types:ip-gateway-type	The IP address of an IP router that can route packets to the destination IP address. The gateway address must be on the same IP subnet as one of the port IP addresses. This leaf is a mandatory leaf and once set, the IP gateway can also be changed.

Name	Type	Description
status-flags	string	The IP route flag: U = Usable G = Gateway H = Host C = Created (Interface) S = Static L = LinkLayer

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<Response>
 <extension-ip-route>
  <name>0/2</name>
  <dp-id>0</dp-id>
  <ip-address>7.9.0.0</ip-address>
  <ip-prefix-length>24</ip-prefix-length>
  <ip-gateway>12.18.20.1</ip-gateway>
  <status-flags>U G S </status-flags>
 </extension-ip-route>
 <extension-ip-route>
  <name>0/2</name>
  <dp-id>0</dp-id>
  <ip-address>12.18.20.0</ip-address>
  <ip-prefix-length>24</ip-prefix-length>
  <ip-gateway>*</ip-gateway>
  <status-flags>U C </status-flags>
 </extension-ip-route>
</Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

```
 <extension-ip-route>
  <name>4/0</name>
  <ip-address>3.0.0.0</ip-address>
  <ip-prefix-length>24</ip-prefix-length>
  <dp-id>0</dp-id>
```

```
</extension-ip-route>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

```
<extension-ip-route>
 <name>4/0</name>
 <ip-address>3.0.0.0</ip-address>
 <ip-prefix-length>24</ip-prefix-length>
 <dp-id>0</dp-id>
 <ip-gateway>2.2.2.3</ip-gateway>
</extension-ip-route>
```

brocade-extension-ipsec-policy/extension-ipsec-policy

Description

Represents IPsec policy defined on extension blade or system.

PATCH related restrictions for config replay:

This list will not return the authentication-data leaf in the GET

request. Hence, the payload from a GET operation response cannot be

subsequently used for PATCH operation. However any config replay with

proper payload values are supported.

Parameters

Table 26: brocade-extension-ipsec-policy/extension-ipsec-policy

Name	Type	Description
policy-name	string	Specifies the name for the IPsec policy. The IPsec policy name can be up to 31 characters long and cannot contain certain special characters and keywords such as 'all' and 'none'.
profile-name	extension-ipsec-profile-name	Specifies the IPsec profile to use with the IPSec policy. Valid values for profile name are 'preshared' and 'pki'.

Name	Type	Description
authentication-data	authentication-data	Specifies the authentication-data depending on the profile used: For 'preshared' profile, this field specifies the preshared-key to be used for authentication. This operand is required with shared-key authentication. Preshared key length should have minimum length of 16 characters and maximum length of 64 characters. For 'pki' profile, this field specifies the local key pair name to use for IKE authentication. This operand is required with PKI profile. Key pair name length should have minimum length of 1 character and maximum length of 32 characters. Authentication data will returned as a zero length string for read operations on policies with 'preshared' profile.
num-ike-sessions	uint16	Number of IKE Sessions.
restart-ike-sessions	uint8	Write-only leaf, will not be returned in GET request. Set it to 1 to restart all inactive IKE sessions for this policy in PATCH request. Please use rpc 'brocade-operation-restart' to restart the IKE sessions for IPsec policies.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <extension-ipsec-policy>
  <policy-name>pki_policy</policy-name>
  <profile-name>pki</profile-name>
  <authentication-data>kprest</authentication-data>
  <num-ike-sessions>0</num-ike-sessions>
  <restart-ike-sessions>0</restart-ike-sessions>
 </extension-ipsec-policy>
 <extension-ipsec-policy>
  <policy-name>preshared_policy</policy-name>
  <profile-name>preshared</profile-name>
  <authentication-data/>
  <num-ike-sessions>0</num-ike-sessions>
  <restart-ike-sessions>0</restart-ike-sessions>
 </extension-ipsec-policy>
</Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

```
<extension-ipsec-policy>
 <policy-name>psk1</policy-name>
 <profile-name>preshared</profile-name>
 <authentication-data>1234567891011121314</authentication-data>
</extension-ipsec-policy>
```

brocade-extension-tunnel/circuit-qos-statistics

Description

Represents circuit statistics for extension blade or system.

Parameters

Table 27: brocade-extension-tunnel/circuit-qos-statistics

Name	Type	Description
ve-port	extension:ve-interface-type	The name of the extension-tunnel interface.
circuit-id	uint16	Circuit identifier the allowed values are for 0 to 9.
ha-type	extension:tunnel-ha-type	Tunnel group HA type configuration.
priority	extension:extension-qos-type	QOS Priority value for the tunnel
flow-status	boolean	Flow control status: false : flow control is off true : flow control is on
operational-status	extension:operational-type	Circuit operational status.
connection-count	yang:zero-based-counter32	Active connection count.
duration	uint32	Connection duration (in seconds).
in-bytes	yang:zero-based-counter64	total octets received.
in-bytes-average	yang:gauge64	Total octets received per 30s average.
in-packets	yang:zero-based-counter64	total packets received.
out-bytes	yang:zero-based-counter64	total octets sent.
out-bytes-average	yang:gauge64	Total octets send per 30s average.
out-packets	yang:zero-based-counter64	Total TCP packets sent.
in-tcp-bytes	yang:zero-based-counter64	Total TCP bytes received.

Name	Type	Description
out-tcp-bytes	yang:zero-based-counter64	Total TCP bytes sent.
in-tcp-packets	yang:zero-based-counter64	TCP packets received.
out-tcp-packets	yang:zero-based-counter64	TCP packets sent.
tcp-retransmits	yang:zero-based-counter64	TCP retransmits /lost packets.
rtt	yang:zero-based-counter64	The round trip time.
tcp-out-of-order-packets	yang:zero-based-counter64	TCP total out of order packets.
tcp-slow-starts	yang:zero-based-counter64	TCP slow starts.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <circuit-qos-statistics>
  <ve-port>8/16</ve-port>
  <circuit-id>1</circuit-id>
  <ha-type>main</ha-type>
  <priority>priority-control</priority>
  <duration>5053</duration>
  <connection-count>1</connection-count>
  <flow-status>false</flow-status>
  <operational-status>2</operational-status>
  <in-bytes>1474356</in-bytes>
  <in-bytes-average>81</in-bytes-average>
  <in-packets>4230</in-packets>
  <out-bytes>1231008</out-bytes>
  <out-bytes-average>176</out-bytes-average>
  <out-packets>4158</out-packets>
  <in-tcp-bytes>11851124</in-tcp-bytes>
  <out-tcp-bytes>14333824</out-tcp-bytes>
  <in-tcp-packets>99545</in-tcp-packets>
  <out-tcp-packets>99544</out-tcp-packets>
  <tcp-retransmits>0</tcp-retransmits>
  <rtt>1</rtt>
  <tcp-out-of-order-packets>0</tcp-out-of-order-packets>
  <tcp-slow-starts>0</tcp-slow-starts>
 </circuit-qos-statistics>
```

```
...
 <circuit-qos-statistics>
  <ve-port>8/17</ve-port>
  <circuit-id>4</circuit-id>
  <ha-type>main</ha-type>
  <priority>IP-priority-low</priority>
  <duration>0</duration>
  <connection-count>0</connection-count>
  <flow-status>true</flow-status>
  <operational-status>12</operational-status>
  <in-bytes>0</in-bytes>
  <in-bytes-average>0</in-bytes-average>
  <in-packets>0</in-packets>
  <out-bytes>0</out-bytes>
  <out-bytes-average>0</out-bytes-average>
  <out-packets>0</out-packets>
  <in-tcp-bytes>0</in-tcp-bytes>
  <out-tcp-bytes>0</out-tcp-bytes>
  <in-tcp-packets>0</in-tcp-packets>
  <out-tcp-packets>0</out-tcp-packets>
  <tcp-retransmits>0</tcp-retransmits>
  <rtt>0</rtt>
  <tcp-out-of-order-packets>0</tcp-out-of-order-packets>
  <tcp-slow-starts>0</tcp-slow-starts>
 </circuit-qos-statistics>
</Response>
```

brocade-extension-tunnel/extension-circuit

Description

Represents circuit interface for extension blade or system.

Parameters

Table 28: brocade-extension-tunnel/extension-circuit

Name	Type	Description
name	extension:ve-interface-type	The name of the interface.
circuit-id	uint16	Circuit identifier. In case of 7810 the range is restricted to 0..5.
local-ip-address	inet:ip-address	Specifies the local IP address to use for the circuit. The default value for IPV4 address is '0.0.0.0'. The default value for IPV6 address is '::'.
remote-ip-address	inet:ip-address	Specifies the remote IP address to use for the circuit. The default value for IPV4 address is '0.0.0.0'. The default value for IPV6 address is '::'.

Name	Type	Description
metric	uint32	Specifies the metric for the configured circuit. A lower metric assigns a higher priority to the circuit. As data is flowing through the extension tunnel, it automatically traverses the lowest metric cost circuits. For example, if a tunnel has four circuits, three of which are set to a metric of 0 and one is set to a metric of 1, all data will flow over the metric 0 circuits. This parameter is meaningful only, if you configure more than one circuit. 0 : Primary 1 : Failover
failover-group-id	uint32	Specifies the failover group ID for the configured circuit. The circuit failover groups must be defined at both ends of the extension tunnel and each failover group should include at least one metric 0 and one metric 1 circuit. If all metric 0 circuits in the failover group go down, the transmission fails over to the metric 1 circuits in the group. If all metric 0 circuits in a tunnel go down, by default the traffic will run over the metric 1 circuits. This field has no meaning when load level for the tunnel is 'spillover'.
admin-enabled	uint8	Is Circuit admin status enabled. Default: 1, Values supported are 0..5 However user is not allowed to set all of them as some of them are Internal RO states. Supported configure values are Disabled and Enabled only. Internal values are Config, HA-disabled, Fenced and Test. Please refer '../is-admin-enabled' leaf as this leaf is deprecated. 0 - Disabled 1 - Enabled 2 - Config 3 - HA-disabled 4 - Fenced 5 - Test. Deprecated: Please use 'is-admin-enabled' leaf instead.
is-admin-enabled	boolean	Is Circuit admin status enabled. Default: true, Values supported are true/false. false - Disabled true - Enabled
admin-status	uint8	Is Circuit admin status enabled. Default: 1, Values supported are 0..5 However user is not allowed to set all of them as some of them are Internal RO states. Supported configure values are Disabled and Enabled only. Internal values are Config, HA-disabled, Fenced and Test. Please refer '../admin-status-v2' leaf as this leaf is deprecated. 0 - Disabled 1 - Enabled 2 - Config 3 - HA-disabled 4 - Fenced 5 - Test.

Name	Type	Description
admin-status-v2	extension:circuit-admin-status-type	Is Circuit admin status enabled. Default: is enabled, However user is not allowed to set all of them as some of them are Internal RO states. Supported configure values are Disabled and Enabled only. Internal values are Config, HA-disabled, Fenced and Test.
path-mtu-discovered	extension:discovered-mtu-type	Discovered Path MTU value for local IP address interface. This field is valid only when path MTU discovery is enabled and the circuit connection with the peer is established. A value of '0' indicates MTU discovery is not enabled on the interface, or the circuit has not yet established a connection to the peer.
ha-path-mtu-discovered	extension:discovered-mtu-type	Discovered Path MTU value for local HA IP address interface. This field is valid only when HA path MTU discovery is enabled, and the HA circuit connection with the peer is established. A value of '0' indicates HA path MTU discovery is not enabled on the interface, or the HA circuit has not yet established a connection to the peer.
minimum-communication-rate	uint32	Minimum Communication Rate of the circuit in kbps. You can set a minimum and a maximum for the committed rate to configure the circuit for Adaptive Rate Limiting (ARL), which allows for a more effective sharing of bandwidth between applications. The maximum committed rate can be no larger than five times the minimum committed rate, and both sides of the circuit must have matching configurations. The range on 7810 is restricted to 0|20000..2500000. The range on 7840 is restricted to 0|20000..10000000. The range on SX6 is restricted to 0|20000..10000000. The range on 7850 is restricted to 0|50000..25000000.
maximum-communication-rate	uint32	Maximum Communication Rate of the circuit in kbps. You can set a minimum and a maximum for the committed rate to configure the circuit for Adaptive Rate Limiting (ARL), which allows for a more effective sharing of bandwidth between applications. The maximum committed rate can be no larger than five times the minimum committed rate, and both sides of the circuit must have matching configurations. The range on 7810 is restricted to 0|20000..2500000.
keep-alive-timeout	uint32	Specifies the keep alive timeout in milliseconds. If the tunnel does not already have FICON Emulation enabled, circuits created on the tunnel default to the the keep alive timeout of 6000 ms (6 seconds). If FICON emulation is enabled on the extension tunnel when a circuit is created, the keep alive timeout defaults to 1000 ms (1 seconds).
vlan-id	extension:vlan-id-type	Specifies the VLAN ID. When not set, this value will show up as 0.
ha-vlan-id	extension:vlan-id-type	Specifies the HA IP address VLAN ID. When not set, this value will show up as 0.
circuit-status	extension:operational-type	Circuit operational status. Please refer '../circuit-status-v2' leaf as this leaf is deprecated.
circuit-status-v2	extension:operational-status-type	Circuit operational status

Name	Type	Description
local-ha-ip-address	inet:ip-address	Specifies the local HA IP address to use for the circuit. This HA configuration is only supported on 7840 and SX6. The default value for IPV4 address is '0.0.0.0'. The default value for IPV6 address is '::'.
remote-ha-ip-address	inet:ip-address	Specifies the remote HA IP address to use for the circuit. The default value for IPV4 address is '0.0.0.0'. The default value for IPV6 address is '::'.
arl-algorithm-mode	uint32	Specifies Adaptive Rate Limiting (ARL) algorithm. The default value in case of IP Extension enabled tunnel is 0. The default value in case of IP Extension disabled tunnel is 1. Please refer '../arl-algorithm-mode-v2' leaf as this leaf is deprecated. Allowable modes are: 0 : auto 1 : static-reset 2 : step-down 3 : timed-step-down
arl-algorithm-mode-v2	extension:arl-algo-type	Specifies Adaptive Rate Limiting (ARL) algorithm. The default value in case of IP Extension enabled tunnel is auto. The default value in case of IP Extension disabled tunnel is static-reset.
l2-cos	Container	L2 class of service/priority as defined by IEEE 802.1p for the tunnel.
- priority-control	string	Specifies the Layer 2 Class of Service (L2CoS) value for F-Class Traffic. This priority setting controls connections between switches.
- fc-priority-high	string	Specifies the L2CoS value for FC High Priority Traffic.
- fc-priority-medium	string	Specifies the L2CoS value for FC Medium Priority Traffic.
- fc-priority-low	string	Specifies the L2CoS value for FC Low Priority Traffic.
- ip-priority-high	string	Specifies the L2CoS value for IP High Priority Traffic.
- ip-priority-medium	string	Specifies the L2CoS value for IP Medium Priority Traffic.
- ip-priority-low	string	Specifies the L2CoS value for IP Low Priority Traffic.
dscp	Container	The Internet Protocol (IP) Differentiated Services Code Point markings for frames.
- priority-control	string	Specifies the DSCP value for F-Class Traffic. The range is 0 to 63. Please refer '../priority-control-v2' leaf as this leaf is deprecated.
- fc-priority-high	string	Specifies the DSCP value for FC High Priority. The range is 0 to 63. Please refer '../fc-priority-high-v2' leaf as this leaf is deprecated.
- fc-priority-medium	string	Specifies the DSCP value for FC Medium Priority. The range is 0 to 63. Please refer '../fc-priority-medium-v2' leaf as this leaf is deprecated.

Name	Type	Description
- fc-priority-low	string	Specifies the DSCP value for FC Low Priority. The range is 0 to 63. Please refer '../fc-priority-low-v2' leaf as this leaf is deprecated.
- ip-priority-high	string	Specifies the DSCP value for IP High Priority. The range is 0 to 63. Please refer '../ip-priority-high-v2' leaf as this leaf is deprecated.
- ip-priority-medium	string	Specifies the DSCP value for IP Medium Priority. The range is 0 to 63. Please refer '../ip-priority-medium-v2' leaf as this leaf is deprecated.
- ip-priority-low	string	Specifies the DSCP value for IP Low Priority. The range is 0 to 63. Please refer '../ip-priority-low-v2' leaf as this leaf is deprecated.
- priority-control-v2	inet:dscp	Specifies the DSCP value for F-Class Traffic. The range is 0 to 63.
- fc-priority-high-v2	inet:dscp	Specifies the DSCP value for FC High Priority. The range is 0 to 63.
- fc-priority-medium-v2	inet:dscp	Specifies the DSCP value for FC Medium Priority. The range is 0 to 63.
- fc-priority-low-v2	inet:dscp	Specifies the DSCP value for FC Low Priority. The range is 0 to 63.
- ip-priority-high-v2	inet:dscp	Specifies the DSCP value for IP High Priority. The range is 0 to 63.
- ip-priority-medium-v2	inet:dscp	Specifies the DSCP value for IP Medium Priority. The range is 0 to 63.
- ip-priority-low-v2	inet:dscp	Specifies the DSCP value for IP Low Priority. The range is 0 to 63.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
{
    "Response": {
        "extension-circuit": [
```

```
            {
                "name": "0/25",
                "circuit-id": 0,
                "local-ip-address": "10.10.10.10",
                "remote-ip-address": "10.10.10.20",
                "metric": 0,
                "admin-enabled": 1,
                "is-admin-enabled": true,
                "admin-status": 1,
                "admin-status-v2": "enabled",
                "path-mtu-discovered": 0,
                "ha-path-mtu-discovered": 0,
                "minimum-communication-rate": 1000000,
                "maximum-communication-rate": 1000000,
                "keep-alive-timeout": 6000,
                "vlan-id": 0,
                "ha-vlan-id": 0,
                "l2-cos": {
                    "priority-control": "0",
                    "fc-priority-high": "0",
                    "fc-priority-medium": "0",
                    "fc-priority-low": "0",
                    "ip-priority-high": "0",
                    "ip-priority-medium": "0",
                    "ip-priority-low": "0"
                },
                "dscp": {
                    "priority-control": "0",
                    "fc-priority-high": "0",
                    "fc-priority-medium": "0",
                    "fc-priority-low": "0",
                    "ip-priority-high": "0",
                    "ip-priority-medium": "0",
                    "ip-priority-low": "0",
                    "priority-control-v2": 0,
                    "fc-priority-high-v2": 0,
                    "fc-priority-medium-v2": 0,
                    "fc-priority-low-v2": 0,
                    "ip-priority-high-v2": 0,
                    "ip-priority-medium-v2": 0,
                    "ip-priority-low-v2": 0
                },
                "failover-group-id": 0,
                "circuit-status": 12,
                "circuit-status-v2": "in-progress",
                "local-ha-ip-address": "10.10.10.100",
                "remote-ha-ip-address": "10.10.10.200",
                "arl-algorithm-mode": 0,
                "arl-algorithm-mode-v2": "auto"
            }
        ]
    }
}
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

```
<extension-circuit>
 <name>4/17</name>
 <circuit-id>0</circuit-id>
</extension-circuit>
```

brocade-extension-tunnel/extension-circuit-statistics

Description

Represents circuit statistics for extension blade or system.

Parameters

Table 29: brocade-extension-tunnel/extension-circuit-statistics

Name	Type	Description
name	extension:ve-interface-type	The name of the interface.
circuit-id	uint16	Circuit identifier.
flow-status	uint16	Flow control status: Please refer '../flow-status-v2' leaf as this leaf is deprecated. 0 : flow control is off 1 : flow control is on
flow-status-v2	boolean	Flow control status: false : flow control is off true : flow control is on
operational-status	extension:operational-type	Circuit operational status.
connection-count	yang:zero-based-counter32	Active connection count.
duration	uint32	Connection duration (in seconds).
out-packet-lost	yang:zero-based-counter64	Number of TCP packets lost because of delay/drop in the Network.
out-packet-total	yang:zero-based-counter64	Number of total TCP packets sent out in the Network.
in-bytes	yang:zero-based-counter64	total octets received.
out-bytes	yang:zero-based-counter64	total octets sent.
rtt	yang:gauge32	The current round trip time.
time-generated	fibrechannel:time-generated-type	The interval based statistics collection time in seconds since EPOCH.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "extension-circuit-statistics": [
            {
                "name": "0/25",
                "circuit-id": 0,
                "flow-status": 1,
                "flow-status-v2": true,
                "operational-status": 12,
                "connection-count": 0,
                "duration": 0,
                "out-packet-total": 0,
                "out-packet-lost": 0,
                "out-bytes": 0,
                "in-bytes": 0,
                "rtt": 0,
                "time-generated": 1677704243
            }
        ]
    }
}
```

brocade-extension-tunnel/extension-tunnel

Description

Represents tunnel interface for extension blade or system.

Parameters

Table 30: brocade-extension-tunnel/extension-tunnel

Name	Type	Description
name	extension:ve-interface-type	The name of the interface.
user-friendly-name	string	Specifies a description for the specified tunnel.
local-wwn	fibrechannel:wwn-type	Specifies the WWN of the local FC entity.

Name	Type	Description
remote-wwn	fibrechannel:wwn-type	WWN that can be optionally configured to be verified against the WWN of the remote FC entity.
peer-wwn	fibrechannel:wwn-type	Specifies the WWN of the remote FC entity.
peer-product-name	fibrechannel:product-name-type	The Peer Extension platform's product name, connected to the remote extension-tunnel endpoint. The peer product name will be seen only once the tunnel has come online and initial extension-tunnel handshake is completed. Note: ===== A value of 'unknown' may be seen if the peer platform is not known for instance if we have a newer Extension platform blade/switch in deployment. We may see this momentarily also for existing tunnels if they are down after cold boot etc. It will also be seen in other case as well where the extension platform is running a Fabric OS version below Fabric OS v9.1.0.
admin-enabled	uint8	Enables or disables the tunnel VE port. Admin status is enabled by default. Please refer '../admin-enabled-v2' as this leaf is deprecated. 0 : Disabled 1 : Enabled
admin-enabled-v2	boolean	Enables or disables the tunnel VE port. Admin status is enabled by default. false : Disabled true : Enabled
fast-write-enabled	uint8	Disables or enables the FCIP FastWrite on the specified extension tunnel. Please refer '../fast-write-enabled-v2' leaf as this leaf is deprecated. NOTE: In order to disable the fast-write-enabled configuration both 'tape-read' and 'tape-write' leaves must also be disabled. 0 : Disable 1 : Enable
fast-write-enabled-v2	boolean	Disables or enables the FCIP FastWrite on the specified extension tunnel. NOTE: In order to disable the fast-write-enabled configuration both 'tape-read-v2' and 'tape-write-v2' leaves must also be disabled. false : Disable true : Enable
tape-read	uint8	Enables read Tape Pipelining (FCIP FastWrite must also be enabled). Read Tape Pipelining can not be enabled if write Tape Pipelining is disabled. Please refer '../tape-read-v2' leaf as this leaf is deprecated. 0 : Disable 1 : Enable

Name	Type	Description
tape-read-v2	boolean	Enables or disables read Tape Pipelining. Read Tape Pipelining can not be enabled when write Tape Pipelining (tape-write-v2) is enabled and FCIP FastWrite (fast-write-enabled-v2) is enabled. false : Disable true : Enable
tape-write	uint8	Enables write Tape Pipelining (FCIP FastWrite must also be enabled). Please refer '../tape-write-v2' leaf as this leaf is deprecated'. 0 : Disable 1 : Enable
tape-write-v2	boolean	Enables or disables write Tape Pipelining. Write Tape Pipelining can be enabled when FCIP FastWrite (fast-write-enabled-v2) is enabled. false : Disable true : Enable
tunnel-status	extension:operational-type	Tunnel operational status. Please refer '../tunnel-status-v2' leaf as this leaf is deprecated.
tunnel-status-v2	extension:operational-status-type	Tunnel operational status.
ipsec-enabled	uint8	Specifies the disabled or enabled state of Internet Protocol Security (IPsec) on the specified tunnel: Please refer '../ipsec-enabled-v2' leaf as this leaf is deprecated. 0 : IPsec is disabled on the tunnel 1 : IPsec is enabled on the tunnel
ipsec-enabled-v2	boolean	Specifies the disabled or enabled state of Internet Protocol Security (IPsec) on the specified tunnel: false : IPsec is disabled on the tunnel true : IPsec is enabled on the tunnel
ipsec-policy	string	Sets the Internet Protocol Security (IPsec) on the specified tunnel to use the specified IPsec Policy, or to disable IPsec for the tunnel if the 'none' operand is used. Policy name can be 1 to 31 characters in length. In case of GET it would return the name of Internet Protocol Security (IPsec) set for the tunnel or else would be an empty string. An empty string value cannot be patched when 'ipsec-policy' is the only leaf in the request.
ha-operational-status	uint16	HA Oper Status of a tunnel: 8 : HA Online 9 : HA Offline 10 : HA Ready
ficon	uint8	Enables or disables FICON emulation on the specified extension tunnel. Please refer '../ficon-v2' leaf as this leaf is deprecated. 0 : Disable 1 : Enable
ficon-v2	boolean	Enables or disables FICON emulation on the specified extension tunnel. false : Disable true : Enable

Name	Type	Description
ficon-xrc-acceleration	uint8	Enables or disables FICON XRC emulation. FICON XRC Emulation allows XRC (IBM eXtendedRemote Copy, also known as IBM z/OS GlobalMirroring) to operate effectively at extended distances. Please refer '../ficon-xrc-acceleration-v2' leaf as this leaf is deprecated. This field is valid only if FICON emulation is enabled on the specified tunnel. 0 : Disable 1 : Enable
ficon-xrc-acceleration-v2	boolean	Enables or disables FICON XRC emulation. FICON XRC Emulation allows XRC (IBM eXtendedRemote Copy, also known as IBM z/OS GlobalMirroring) to operate effectively at extended distances. This field is valid only if FICON emulation is enabled on the specified tunnel. false : Disable true : Enable
ficon-tape-write-acceleration	uint8	Enables or disables FICON Tape Write Pipelining. This feature improves the performance of certain applications when writing to tape over extended distances. Please refer '../ficon-tape-write-acceleration-v2' leaf as this leaf is deprecated. This field is valid only if FICON emulation is enabled on the specified tunnel. 0 : Disable 1 : Enable
ficon-tape-write-acceleration-v2	boolean	Enables or disables FICON Tape Write Pipelining. This feature improves the performance of certain applications when writing to tape over extended distances. This field is valid only if FICON emulation is enabled on the specified tunnel. false : Disable true : Enable
ficon-tape-read-acceleration	uint8	Enables or disables FICON Tape Read Pipelining. This feature improves performance for certain applications when reading from FICON tape over extended distances. Please refer 'ficon-tape-read-acceleration-v2' leaf as this leaf is deprecated. This field is valid only if FICON emulation is enabled on the specified tunnel. 0 : Disable 1 : Enable

Name	Type	Description
ficon-tape-read-acceleration-v2	boolean	Enables or disables FICON Tape Read Pipelining. This feature improves performance for certain applications when reading from FICON tape over extended distances. This field is valid only if FICON emulation is enabled on the specified tunnel. false : Disable true : Enable
ficon-tin-tir-emulation	uint8	Enables or disables FICON TIN/TIR emulation. This feature enhances recovery when a TIN/TIR exchange occurs as part of a channel recovery operation during tape emulation. This feature is enabled by default (recommended setting). Please refer '../ficon-tin-tir-emulation-v2' leaf as this leaf is deprecated. This field is valid only if FICON emulation is enabled on the specified tunnel. 0 : Disable 1 : Enable
ficon-tin-tir-emulation-v2	boolean	Enables or disables FICON TIN/TIR emulation. This feature enhances recovery when a TIN/TIR exchange occurs as part of a channel recovery operation during tape emulation. This feature is enabled by default (recommended setting). This field is valid only if FICON emulation is enabled on the specified tunnel. false : Disable true : Enable
ficon-device-acknowledgement-emulation	uint8	Enables or disables FICON Device Level Acknowledgement emulation. This feature is applicable to both FICON Disk and Tape configurations. The feature removes one network round trip for exchanges that end with a Device Level Acknowledgement frame from the device. This feature is enabled by default (recommended setting). Please refer '../ficon-device-acknowledgement-emulation-v2' leaf as this leaf is deprecated. This field is valid only if FICON emulation is enabled on the specified tunnel. 0 : Disable 1 : Enable
ficon-device-acknowledgement-emulation-v2	boolean	Enables or disables FICON Device Level Acknowledgement emulation. This feature is applicable to both FICON Disk and Tape configurations. The feature removes one network round trip for exchanges that end with a Device Level Acknowledgement frame from the device. This feature is enabled by default (recommended setting). This field is valid only if FICON emulation is enabled on the specified tunnel. false : Disable true : Enable

Name	Type	Description
ficon-read-block-id-emulation	uint8	Enables or disables FICON read Tape Read Block ID emulation. This feature permits FICON write channel programs containing embedded read block ID commands (CCWs) with a byte count of exactly four bytes to be processed as emulated commands during write emulation processes. Please refer 'ficon-read-block-id-emulation-v2' leaf as this leaf is deprecated. This field is valid only if FICON emulation is enabled on the specified tunnel. 0 : Disable 1 : Enable
ficon-read-block-id-emulation-v2	boolean	Enables or disables FICON read Tape Read Block ID emulation. This feature permits FICON write channel programs containing embedded read block ID commands (CCWs) with a byte count of exactly four bytes to be processed as emulated commands during write emulation processes. This field is valid only if FICON emulation is enabled on the specified tunnel. false : Disable true : Enable
ficon-teradata-read-acceleration	uint8	Enables or disables FICON Read Emulation for a Teradata server on the specified extension tunnel. Please refer 'ficon-teradata-read-acceleration-v2' leaf as this leaf is deprecated. This field is valid only if FICON emulation is enabled on the specified tunnel. 0 : Disable 1 : Enable
ficon-teradata-read-acceleration-v2	boolean	Enables or disables FICON Read Emulation for a Teradata server on the specified extension tunnel. This field is valid only if FICON emulation is enabled on the specified tunnel. false : Disable true : Enable
ficon-teradata-write-acceleration	uint8	Enables or disables FICON Write Emulation for a Teradata server on the specified extension tunnel. Please refer '../ficon-teradata-write-acceleration-v2' leaf as this leaf is deprecated. This field is valid only if FICON emulation is enabled on the specified tunnel. 0 : Disable 1 : Enable
ficon-teradata-write-acceleration-v2	boolean	Enables or disables FICON Write Emulation for a Teradata server on the specified extension tunnel. This field is valid only if FICON emulation is enabled on the specified tunnel. false : Disable true : Enable

Name	Type	Description
ficon-tape-write-max-pipe	uint32	Defines the maximum number of tape write channel commands (CCWs) that can enter the write pipeline for a single device whether all the CCWs are bundled in a single channel program or in multiple channel programs. The setting has significance only for host (channel) initiated operations at this side and will not affect tape write operations initiated by hosts (channels) attached at the opposite side. Too small of a value will result in poor performance. The value should be chosen based upon the typical tape channel program that requires optimum performance. This field is valid only if FICON emulation is enabled on the specified tunnel.
ficon-tape-read-max-pipe	uint32	Defines the maximum number of tape read channel commands (CCWs) that can enter the read pipeline for a single device whether all the CCWs are bundled in a single channel program or in multiple channel programs. The setting has significance only for host (channel) initiated operations at this side and will not affect tape write operations initiated by hosts (channels) attached at the opposite side. Too small of a value will result in poor performance. The value should be chosen based upon the typical tape channel program that requires optimum performance. This field is valid only if FICON emulation is enabled on the specified tunnel.
ficon-tape-write-max-devices	uint32	Defines the maximum number of concurrent emulated tape write operations. As concurrency increases, the value of emulation decreases. Excessive concurrency has the potential to oversubscribe packet data memory. The setting has significance only for host (channel) initiated operations at this side and will not affect tape write operations initiated by hosts (channels) attached. This field is valid only if FICON emulation is enabled on the specified tunnel.
ficon-tape-read-max-devices	uint32	Defines the maximum number of concurrent emulated tape read operations. As concurrency increases, the value of emulation decreases. Excessive concurrency has the potential to oversubscribe packet data memory. The setting has significance only for host (channel) initiated operations at this side and will not affect tape read operations initiated by hosts (channels) attached at the opposite side. This field is valid only if FICON emulation is enabled on the specified tunnel.
ficon-tape-write-timer	uint32	Defines a time limit for pipelined write chains. This value is specified in milliseconds (ms). If a pipelined write chain takes longer than this value to complete, the ending status for the next write chain will be withheld from the channel. This limits processing to what the network and device can support. Too small a value limits pipelining performance. Too large a value results in too much data being accepted for one device on a path. This field is valid only if FICON emulation is enabled on the specified tunnel.

Name	Type	Description
ficon-tape-write-max-chain	uint32	Defines the maximum amount of data that can be contained in a single CCW chain in bytes. If this value is exceeded, emulation is suspended. This field is valid only if FICON emulation is enabled on the specified tunnel.
ficon-oxid-base	uint32	Defines the base value of an entry pool of 256 OXIDs supplied to emulation-generated exchanges. It should fall outside the range used by FICON channels and devices to avoid conflicts. The default value is 0xB000 (recommended setting). This field is valid only if FICON emulation is enabled on the specified tunnel.
ip-extension	uint16	Enables or disables IP Extension capability of a tunnel. Please refer '../ip-extension-v2' leaf as this leaf is deprecated. Possible values: 0 : Disable 1 : Enable
ip-extension-v2	boolean	Enables or disables IP Extension capability of a tunnel. Possible values: false : Disable true : Enable
load-level	extension:tunnel-load-level	Specifies spillover or failover load-balancing method. The default load-balancing method is 'failover'. Possible values are: 'default', 'failover' and 'spillover'
active-load-level	extension:tunnel-load-level	Load level value currently active on the tunnel.
peer-load-level	extension:tunnel-load-level	Specifies the load level configured on the peer-end of the tunnel.
compression-tunnel	uint16	Configures compression on the specified tunnel. The value (5 : Fast Deflate) is only supported on SX6 and 7840. Please refer '../compression-tunnel-v2' leaf as this leaf is deprecated. 0 : Off (Compression disabled) 3 : Deflate 4 : Aggressive Deflate 5 : Fast Deflate
compression-tunnel-v2	extension:compression-tunnel-type	Configures compression on the specified tunnel.
compression-protocol	Container	This property denotes the FC/IP compression mode for the tunnel.

Name	Type	Description
- fc-compression	string	Specifies FC compression on the specified tunnel. It will return -1 when default or otherwise the configured compression set for FC protocol. The value (5 : Fast Deflate) is only supported on SX6 and 7840. Please refer '../fc-compression-v2' leaf as this leaf is deprecated. Specifies one of the following values for compression_level: -1 : Default (Enable default compression level specified in compression-tunnel attribute) 0 : Off (Compression disabled) 3 : Deflate 4 : Aggressive Deflate 5 : Fast Deflate
- fc-compression-v2	extension:compression-protocol-type	Specifies FC compression on the specified tunnel. It will return 'default' when default or otherwise the configured compression set for FC protocol. The value (Fast Deflate) is only supported on SX6 and 7840.
- ip-compression	string	Specifies IP compression on the specified tunnel. It will return -1 when default or otherwise the configured compression set for IP protocol. Please refer '../ip-compression-v2' leaf as this leaf is deprecated. Specifies one of the following values for compression_level: -1 : Default (Enable default compression level specified in compression-tunnel attribute) 0 : Off (Compression disabled) 3 : Deflate 4 : Aggressive Deflate
- ip-compression-v2	extension:compression-protocol-type	Specifies IP compression on the specified tunnel. It will return 'default' when default or otherwise the configured compression set for IP protocol.
qos-ratio	Container	Specifies tunnel bandwidth distribution values.
- distribution	string	Specifies tunnel bandwidth distribution mode: 1 : Default, for FCIP mode 2 : Protocol, for Hybrid mode The default value for IP Extension enabled tunnel is 2. The default value for IP Extension disabled tunnel is 1.
- distribution-value	string	Specifies values for bandwidth distribution: '<Percentage-for-FC-bandwidth>,<Percentage-for-IP-bandwidth>' - for Hybrid mode. The total percentage distribution for FCIP and IP extension must always add up to 100%. For example, the value in case of Hybrid mode by default is '50,50' which is 50% each for FCIP and IP extension protocol respectively. The minimum distribution percentage allowed for each protocol is 10%.
- fc-high-qos	string	Specifies QoS percentage for FC high priority. Each priority can have a minimum of 10%. The sum of percentages must equal 100%.

Name	Type	Description
- fc-medium-qos	string	Specifies QoS percentage for FC medium priority. Each priority can have a minimum of 10%. The sum of percentages must equal 100%.
- fc-low-qos	string	Specifies QoS percentage for FC low priority. Each priority can have a minimum of 10%. The sum of percentages must equal 100%.
- ip-high-qos	string	Specifies QoS percentage for IP high priority. Each priority can have a minimum of 10%. The sum of percentages must equal 100%. This field is valid only if IP extension is enabled on the tunnel.
- ip-medium-qos	string	Specifies QoS percentage for IP medium priority. Each priority can have a minimum of 10%. The sum of percentages must equal 100%. This field is valid only if IP extension is enabled on the tunnel.
- ip-low-qos	string	Specifies QoS percentage for IP low priority. Each priority can have a minimum of 10%. The sum of percentages must equal 100%. This field is valid only if IP extension is enabled on the tunnel.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "extension-tunnel": [
            {
                "name": "0/25",
                "user-friendly-name": "",
                "local-wwn": "10:00:c4:f5:7c:02:8b:08",
                "remote-wwn": "00:00:00:00:00:00:00:00",
                "peer-wwn": "00:00:00:00:00:00:00:00",
                "peer-product-name": "unknown",
                "compression-tunnel": 0,
                "compression-tunnel-v2": "disabled",
                "admin-enabled": 1,
                "admin-enabled-v2": true,
                "fast-write-enabled": 0,
                "fast-write-enabled-v2": false,
                "tape-read": 0,
                "tape-read-v2": false,
                "tape-write": 0,
                "tape-write-v2": false,
                "tunnel-status": 12,
```

```
                "tunnel-status-v2": "in-progress",
                "ipsec-enabled": 0,
                "ipsec-enabled-v2": false,
                "ipsec-policy": "",
                "ha-operational-status": 10,
                "ficon": 0,
                "ficon-v2": false,
                "ip-extension": 0,
                "ip-extension-v2": false,
                "qos-ratio": {
                    "distribution": "1",
                    "fc-high-qos": "50",
                    "fc-medium-qos": "30",
                    "fc-low-qos": "20"
                },
                "load-level": "failover",
                "active-load-level": "failover",
                "peer-load-level": "failover"
            }
        ]
    }
}
```


DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.


```
<extension-tunnel>
 <name>4/17</name>
</extension-tunnel>
```


brocade-extension-tunnel/extension-tunnel-statistics

Description

Represents tunnel statistics for extension blade or system.

Parameters

Table 31: brocade-extension-tunnel/extension-tunnel-statistics

Name	Type	Description
name	extension:ve-interface-type	The name of the interface.
flow-status	uint16	Flow control status: Please refer '../flow-status-v2' leaf as this leaf is deprecated. 0 : flow control is off 1 : flow control is on
flow-status-v2	boolean	Flow control status: false : flow control is off true : flow control is on
operational-status	extension:operational-type	Tunnel operational status.
connection-count	yang:zero-based-counter32	Active connection count.
duration	uint32	Connection duration (in seconds).
uncompressed-bytes	yang:zero-based-counter64	Uncompressed Bytes.
compressed-bytes	yang:zero-based-counter64	Compressed Bytes.
local-hcl-in-progress	boolean	Indicates whether local HCL is in progress. true: local HCL is in progress false: local HCL is not in progress.
remote-hcl-in-progress	boolean	Indicates whether remote HCL is in progress true: Remote HCL in progress. false: No remote HCL in progress.
fc-ha-status	extension:operational-status-type	The FC HA status of the extension-tunnel.
ip-ha-status	extension:operational-status-type	The IP HA status of the extension-tunnel.
last-error	uint32	The last error encountered during HCL, for an extension-tunnel. The error will be cleared as part of VE recovery sequence after HCL processing is complete. This leaf is seen conditionally only when there is an error otherwise this leaf will be skipped in the GET response.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "extension-tunnel-statistics": [
            {
                "name": "0/25",
                "flow-status": 0,
                "flow-status-v2": false,
                "operational-status": 12,
                "connection-count": 0,
                "duration": 0,
                "uncompressed-bytes": 0,
                "compressed-bytes": 0,
                "local-hcl-in-progress": false,
                "remote-hcl-in-progress": false,
                "fc-ha-status": "HA-ready",
                "ip-ha-status": "disabled"
            }
        ]
    }
}
```


brocade-extension-tunnel/wan-statistics-v1

Description

Represents TCP WAN statistics for extension blade or system.

Parameters

Table 32: brocade-extension-tunnel/wan-statistics-v1

Name	Type	Description
ve-port	extension:ve-interface-type	The name of the interface.
circuit-id	uint16	Circuit identifier.
connection-id	uint32	TCP connection identifier associated with the WAN TCP connection. The TCP connection identifier may change in case of connection going down or establishment. These are not persistent identifiers associated with the connection.
ha-type	extension:tunnel-ha-type	Tunnel HA type configuration.
priority	extension:extension-qos-type	QOS Priority value for the tunnel
source-port	inet:port-number	TCP source port number
destination-port	inet:port-number	TCP remote destination port number
connection-mss	yang:zero-based-counter32	TCP connection MSS.
arl-minimum	yang:gauge32	ARL minimum value
arl-maximum	yang:gauge32	ARL maximum value
arl-current	yang:gauge32	ARL current value

Name	Type	Description
arl-next-reset-algorithm	extension:arl-algo-type	ARL next reset algorithm.
out-bytes	yang:zero-based-counter64	Total bytes sent.
out-packets	yang:zero-based-counter64	Total TCP packets sent.
rtt	yang:gauge32	round trip time.
rtt-maximum	yang:gauge32	maximum round trip time.
rtt-variance	yang:gauge32	variance in round trip time.
rtt-variance-maximum	yang:gauge32	Maximum variance in round trip time.
out-window-size	yang:gauge32	Source window size.
out-window-scale	uint32	Source window scaling shift count.
slow-start-threshold	yang:gauge32	Source slow start threshold.
congestion-window	yang:gauge32	The congestion window is the maximum data that can be sent before an ACK is received.
out-queued-packets-next-sequence	yang:zero-based-counter32	Source TCP queued packets next sequence number.
out-queued-packets-minimum-sequence	yang:zero-based-counter32	Source TCP queued packets minimum sequence number.
out-queued-packets-maximum-sequence	yang:zero-based-counter32	Source TCP queued packets maximum sequence number.
out-in-flight-packets	yang:gauge32	Source in flight packets.
out-unacknowledged-packets-sequence	yang:zero-based-counter32	Source unacknowledged packets sequence.
fast-retransmits	yang:zero-based-counter64	TCP fast retransmits
maximum-fast-retransmits	yang:zero-based-counter64	TCP maximum fast retransmits
slow-retransmits	yang:zero-based-counter64	TCP slow retransmits
duplicate-acknowledgement	yang:zero-based-counter64	TCP duplicate acknowledgement received.
slow-starts	yang:zero-based-counter64	TCP slow starts.
in-bytes	yang:zero-based-counter64	Total bytes received.
in-packets	yang:zero-based-counter64	Total TCP packets received.
in-window-size	yang:zero-based-counter32	Receiver window size.
in-window-size-maximum	yang:zero-based-counter32	Receiver maximum window size.
in-window-scale	uint32	Receiver negotiated window scale.
in-queued-packets	yang:gauge32	TCP queued packets received.
in-queued-packets-next-sequence	yang:zero-based-counter32	TCP queued packets next sequence number received.
in-queued-packets-minimum-sequence	yang:zero-based-counter32	TCP queued packets minimum sequence number received.
in-queued-packets-maximum-sequence	yang:zero-based-counter32	TCP queued packets maximum sequence number received.
in-queued-out-of-order	yang:zero-based-counter32	TCP queued out of order packets received.

Name	Type	Description
in-queued-out-of-order-maximum	yang:zero-based-counter32	TCP total out of order packets received.

Produces

- application/xml
- application/json

brocade-extension/dp-hcl-status

Description

Represents the HCL status on extension datapath process.

Parameters

Table 33: brocade-extension/dp-hcl-status

Name	Type	Description
slot	uint16	The slot number of for the datapath processor.
dp-id	extension:dp-id-type	Extension Data Path Processor ID. Based on platform either it will have a single DP or dual DP. In case of single DP only DP0 is supported, and in case of dual DP both DP0 and DP1 are supported 0 : DP0 1 : DP1.
state	extension:dp-state-type	The current DP HCL state
status	extension:dp-actstate-type	The current DP status.
fc-hcl-stage	extension:dp-hcl-stage-type	The current DP HCL stage for FC protocol.
ip-hcl-stage	extension:dp-hcl-stage-type	The current DP HCL stage for IP protocol.
svi-swapped	boolean	Is the SVI swapped for HCL processing
dp-communication-status	extension:dp-dp-communication-type	The current state of DP-DP communication
firmware-version	string	A human readable string identifying the firmware version running on the datapath process of the switch/blade.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-extension/global-lan-statistics

Description

Represents global LAN DP statistics for extension blade or system.

Parameters

Table 34: brocade-extension/global-lan-statistics

Name	Type	Description
slot	uint16	In case of non-chassis system, the slot number is always 0. In case of chassis system, it is the slot number of chassis in which the extension blade is inserted in. In case of chassis, slot number is non-zero value.
dp-id	extension:dp-id-type	Extension Data Path Processor ID. Based on platform either it will have a single DP or dual DP. In case of single DP only DP0 is supported, and in case of dual DP both DP0 and DP1 are supported 0 : DP0 1 : DP1.
active-tcp-connections	yang:gauge32	Active TCP connection count.
active-tcp-connections-on-local-backup	yang:gauge32	Active clone HCL TCP connection count on Local Backup tunnel.
active-tcp-connections-on-remote-backup	yang:gauge32	Active clone HCL TCP connection count on Remote Backup tunnel.
established-tcp-connections	yang:gauge32	Total LAN TCP connections established.
closed-tcp-connections	yang:gauge32	Total LAN TCP connections closed.
out-tcp-bytes	yang:zero-based-counter64	Total number of TCP bytes transmitted.
in-tcp-bytes	yang:zero-based-counter64	Total number of TCP bytes received.
tcp-tcl-deny-connections	yang:zero-based-counter64	Number of TCP connection denied based on TCL deny rule.
tcp-tcl-lookup-fail	yang:zero-based-counter64	Number of TCP connection denied based on TCL lookup failure.
syn-received	yang:zero-based-counter64	Number of SYN packets received.
syn-fail	yang:zero-based-counter64	Number of SYN packets dropped due to error.
drop-bytes	yang:zero-based-counter64	LSM Tx TCP drop bytes.
drop-packets	yang:zero-based-counter64	Number of LSM packet dropped in egress.
stale-reset-from-host	yang:zero-based-counter64	Stale reset from host.
maximum-tcp-connection-exceeded-as-client	yang:zero-based-counter64	Maximum connection exceeded on listen connection allocation.

Name	Type	Description
maximum-tcp-connection-exceeded-as-server	yang:zero-based-counter64	Maximum connection exceeded during active connection allocation.
maximum-tcp-connection-per-second-exceeded-as-client	yang:zero-based-counter64	Maximum connection per second exceeded on listen connection allocation.
maximum-tcp-connection-per-second-exceeded-as-server	yang:zero-based-counter64	Maximum connection per second exceeded on listen connection allocation.
maximum-udp-connection-exceeded	yang:zero-based-counter64	Total UDP flows failed due to Maximum context on ingress and egress.
maximum-udp-connection-exceeded-on-egress	yang:zero-based-counter64	Total UDP flows failed due to Maximum context on egress.
udp-packets-sent-as-is	yang:zero-based-counter64	Total UDP packets sent as-is.
out-pdu-preserve-on	yang:zero-based-counter64	Number of times transmit with PDU preserve ON.
in-pdu-preserve-on	yang:zero-based-counter64	Number of packets received with PDU preserve ON.
total-ipv6-packets	yang:zero-based-counter64	Total IPv6 packets received - IPv6 WQEs received by IP API layer.
flow-control-on	yang:zero-based-counter64	Flow control on from FTNL.
flow-control-off	yang:zero-based-counter64	Flow control off from FTNL.
active-udp-connections	yang:zero-based-counter64	Number of active NT UDP flows.
established-udp-connections	yang:zero-based-counter64	Number of UDP connections opened since bootup.
closed-udp-connections	yang:zero-based-counter64	Number of UDP connections closed since bootup.
udp-route-lookup-fail	yang:zero-based-counter64	Drop of NT PDUs on egress due to route lookup failure.
udp-pdu-drops-due-to-pko-flow-control	yang:zero-based-counter64	Drop of NT PDUs on egress due to PKO flow control.
out-udp-pdus	yang:zero-based-counter64	Number of Tx UDP PDU.
out-udp-pdu-drops	yang:zero-based-counter64	Number of Tx UDP PDUs dropped due to different reasons.
in-udp-pdus	yang:zero-based-counter64	Number of UDP PDUs received from host.
in-udp-tcl-lookup-fail-pdus	yang:zero-based-counter64	Number of UDP PDUs dropped due to TCL lookup.
in-udp-tcl-deny-pdus	yang:zero-based-counter64	Number of UDP PDUs dropped due to TCL deny.
in-total-udp-pdu-drops	yang:zero-based-counter64	Total UDP PDUs dropped due to multiple reasons.
in-udp-pdu-drops-due-to-stream-flow-control	yang:zero-based-counter64	Number of UDP PDUs dropped due to stream flow control.
out-icmp-pdus	yang:zero-based-counter64	Number of Tx ICMP PDUs.
out-icmp-pdu-drops	yang:zero-based-counter64	Number of Tx ICMP PDU dropped.
in-icmp-pdus	yang:zero-based-counter64	Number of ICMP PDUs received.
in-icmp-tcl-lookup-fail-pdus	yang:zero-based-counter64	Number of ICMP PDUs dropped due to TCL lookup failure.
in-icmp-tcl-deny-pdus	yang:zero-based-counter64	Number of ICMP PDUs dropped due to TCL returning deny status.
in-total-icmp-pdu-drops	yang:zero-based-counter64	Number of ICMP PDU drops.

Name	Type	Description
in-icmp-pdu-drops-due-to-stream-flow-control	yang:zero-based-counter64	Number of ICMP PDU drop due to stream flow control.
out-as-is-ip-pdus	yang:zero-based-counter64	Number of as-is Tx IP PDUs.
out-as-is-ip-pdu-drops	yang:zero-based-counter64	Number of as-is Tx IP PDUs dropped.
in-as-is-ip-pdus	yang:zero-based-counter64	non-terminated TCP, non-batched UDP, Non-ICMP PDU received.
in-as-is-ip-tcl-lookup-fail-pdus	yang:zero-based-counter64	non-terminated TCP, non-batched UDP, Non-ICMP PDU dropped due to TCL lookup failure.
in-as-is-ip-tcl-deny-pdus	yang:zero-based-counter64	non-terminated TCP, non-batched UDP, Non-ICMP PDU dropped due to TCL deny status.
in-total-as-is-ip-pdu-drops	yang:zero-based-counter64	Total non-terminated TCP, non-batched UDP, Non-ICMP PDU dropped in ingress.
in-as-is-ip-pdu-drops-due-to-stream-flow-control	yang:zero-based-counter64	non-terminated TCP, non-batched UDP, Non-ICMP PDU dropped due to stream flow control.
in-error-ip-checksum	yang:zero-based-counter64	Number of IP checksum errors detected on Rx packets.
in-error-tcp-checksum	yang:zero-based-counter64	Number of TCP checksum errors detected on Rx packets.
in-error-mac	yang:zero-based-counter64	Number of MAC errors detected on Rx packets.
in-error-crc	yang:zero-based-counter64	Number of CRC errors detected on Rx packets.
in-error-parity	yang:zero-based-counter64	Number of Parity errors detected on Rx packets.
in-error-length	yang:zero-based-counter64	Number of Length errors detected on Rx packets.
out-udp-packets-64-bytes	yang:zero-based-counter64	Number of UDP packets transmitted of size less than 64 bytes.
out-udp-packets-128-bytes	yang:zero-based-counter64	Number of UDP packets transmitted of size equal to & greater than 64 bytes but less than 128 bytes.
out-udp-packets-256-bytes	yang:zero-based-counter64	Number of UDP packets transmitted of size equal to & greater than 128 bytes but less than 256 bytes.
out-udp-packets-512-bytes	yang:zero-based-counter64	Number of UDP packets transmitted of size equal to & greater than 256 bytes but less than 512 bytes.
out-udp-packets-1024-bytes	yang:zero-based-counter64	Number of UDP packets transmitted of size equal to & greater than 512 bytes but less than 1024 bytes.
out-udp-packets-1500-bytes	yang:zero-based-counter64	Number of UDP packets transmitted of size equal to & greater than 1024 bytes but less than 1500 bytes.
out-udp-packets-3000-bytes	yang:zero-based-counter64	Number of UDP packets transmitted of size equal to & greater than 1500 bytes but less than 3000 bytes.
out-udp-packets-4500-bytes	yang:zero-based-counter64	Number of UDP packets transmitted of size equal to & greater than 3000 bytes but less than 4500 bytes.
out-udp-packets-6000-bytes	yang:zero-based-counter64	Number of UDP packets transmitted of size equal to & greater than 4500 bytes but less than 6000 bytes.
out-udp-packets-9000-bytes	yang:zero-based-counter64	Number of UDP packets transmitted of size equal to & greater than 6000 bytes but less than 9000 bytes.
in-udp-packets-64-bytes	yang:zero-based-counter64	Number of UDP packets received of size less than 64 bytes.

Name	Type	Description
in-udp-packets-128-bytes	yang:zero-based-counter64	Number of UDP packets received of size equal to & greater than 64 bytes but less than 128 bytes.
in-udp-packets-256-bytes	yang:zero-based-counter64	Number of UDP packets received of size equal to & greater than 128 bytes but less than 256 bytes.
in-udp-packets-512-bytes	yang:zero-based-counter64	Number of UDP packets received of size equal to & greater than 256 bytes but less than 512 bytes.
in-udp-packets-1024-bytes	yang:zero-based-counter64	Number of UDP packets received of size equal to & greater than 512 bytes but less than 1024 bytes.
in-udp-packets-1500-bytes	yang:zero-based-counter64	Number of UDP packets received of size equal to & greater than 1024 bytes but less than 1500 bytes.
in-udp-packets-3000-bytes	yang:zero-based-counter64	Number of UDP packets received of size equal to & greater than 1500 bytes but less than 3000 bytes.
in-udp-packets-4500-bytes	yang:zero-based-counter64	Number of UDP packets received of size equal to & greater than 3000 bytes but less than 4500 bytes.
in-udp-packets-6000-bytes	yang:zero-based-counter64	Number of UDP packets received of size equal to & greater than 4500 bytes but less than 6000 bytes.
in-udp-packets-9000-bytes	yang:zero-based-counter64	Number of UDP packets received of size equal to & greater than 6000 bytes but less than 9000 bytes.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<Response>
 <global-lan-statistics>
  <slot>8</slot>
  <dp-id>0</dp-id>
  <active-tcp-connections>0</active-tcp-connections>
  <active-tcp-connections-on-local-backup>0</active-tcp-connections-on-local-backup>
  <active-tcp-connections-on-remote-backup>0</active-tcp-connections-on-remote-backup>
  <established-tcp-connections>0</established-tcp-connections>
  <closed-tcp-connections>0</closed-tcp-connections>
  <out-tcp-bytes>0</out-tcp-bytes>
  <in-tcp-bytes>0</in-tcp-bytes>
  <tcp-tcl-deny-connections>0</tcp-tcl-deny-connections>
  <tcp-tcl-lookup-fail>0</tcp-tcl-lookup-fail>
  <syn-received>0</syn-received>
  <syn-fail>0</syn-fail>
```

```
  <drop-bytes>0</drop-bytes>
  <drop-packets>0</drop-packets>
  <stale-reset-from-host>0</stale-reset-from-host>
  <maximum-tcp-connection-exceeded-as-client>0</maximum-tcp-connection-exceeded-as-client>
  <maximum-tcp-connection-exceeded-as-server>0</maximum-tcp-connection-exceeded-as-server>
  <maximum-tcp-connection-per-second-exceeded-as-client>0</maximum-tcp-connection-per-second-exceededas-
client>
  <maximum-tcp-connection-per-second-exceeded-as-server>0</maximum-tcp-connection-per-second-exceededas-
server>
  <maximum-udp-connection-exceeded>0</maximum-udp-connection-exceeded>
  <maximum-udp-connection-exceeded-on-egress>0</maximum-udp-connection-exceeded-on-egress>
  <udp-packets-sent-as-is>0</udp-packets-sent-as-is>
  <out-pdu-preserve-on>0</out-pdu-preserve-on>
  <in-pdu-preserve-on>0</in-pdu-preserve-on>
  <total-ipv6-packets>12829</total-ipv6-packets>
  <flow-control-on>0</flow-control-on>
  <flow-control-off>0</flow-control-off>
  <active-udp-connections>0</active-udp-connections>
  <established-udp-connections>11643</established-udp-connections>
  <closed-udp-connections>11643</closed-udp-connections>
  <udp-route-lookup-fail>0</udp-route-lookup-fail>
  <udp-pdu-drops-due-to-pko-flow-control>0</udp-pdu-drops-due-to-pko-flow-control>
  <out-udp-pdus>0</out-udp-pdus>
  <out-udp-pdu-drops>0</out-udp-pdu-drops>
  <in-udp-pdus>20549</in-udp-pdus>
  <in-udp-tcl-lookup-fail-pdus>15</in-udp-tcl-lookup-fail-pdus>
  <in-udp-tcl-deny-pdus>1036</in-udp-tcl-deny-pdus>
  <in-total-udp-pdu-drops>1077</in-total-udp-pdu-drops>
  <in-udp-pdu-drops-due-to-stream-flow-control>0</in-udp-pdu-drops-due-to-stream-flow-control>
  <out-icmp-pdus>0</out-icmp-pdus>
  <out-icmp-pdu-drops>0</out-icmp-pdu-drops>
  <in-icmp-pdus>0</in-icmp-pdus>
  <in-icmp-tcl-lookup-fail-pdus>0</in-icmp-tcl-lookup-fail-pdus>
  <in-icmp-tcl-deny-pdus>0</in-icmp-tcl-deny-pdus>
  <in-total-icmp-pdu-drops>0</in-total-icmp-pdu-drops>
  <in-icmp-pdu-drops-due-to-stream-flow-control>0</in-icmp-pdu-drops-due-to-stream-flow-control>
  <out-as-is-ip-pdus>0</out-as-is-ip-pdus>
  <out-as-is-ip-pdu-drops>0</out-as-is-ip-pdu-drops>
  <in-as-is-ip-pdus>1764</in-as-is-ip-pdus>
  <in-as-is-ip-tcl-lookup-fail-pdus>12</in-as-is-ip-tcl-lookup-fail-pdus>
  <in-as-is-ip-tcl-deny-pdus>84</in-as-is-ip-tcl-deny-pdus>
  <in-total-as-is-ip-pdu-drops>96</in-total-as-is-ip-pdu-drops>
  <in-as-is-ip-pdu-drops-due-to-stream-flow-control>0</in-as-is-ip-pdu-drops-due-to-stream-flow-control>
  <in-error-ip-checksum>0</in-error-ip-checksum>
  <in-error-tcp-checksum>0</in-error-tcp-checksum>
  <in-error-mac>0</in-error-mac>
  <in-error-crc>0</in-error-crc>
  <in-error-parity>0</in-error-parity>
  <in-error-length>0</in-error-length>
  <out-udp-packets-64-bytes>0</out-udp-packets-64-bytes>
  <out-udp-packets-128-bytes>0</out-udp-packets-128-bytes>
  <out-udp-packets-256-bytes>0</out-udp-packets-256-bytes>
  <out-udp-packets-512-bytes>0</out-udp-packets-512-bytes>
```

```
   <out-udp-packets-1024-bytes>0</out-udp-packets-1024-bytes>
   <out-udp-packets-1500-bytes>0</out-udp-packets-1500-bytes>
   <out-udp-packets-3000-bytes>0</out-udp-packets-3000-bytes>
   <out-udp-packets-4500-bytes>0</out-udp-packets-4500-bytes>
   <out-udp-packets-6000-bytes>0</out-udp-packets-6000-bytes>
   <out-udp-packets-9000-bytes>0</out-udp-packets-9000-bytes>
   <in-udp-packets-64-bytes>6263</in-udp-packets-64-bytes>
   <in-udp-packets-128-bytes>9866</in-udp-packets-128-bytes>
   <in-udp-packets-256-bytes>880</in-udp-packets-256-bytes>
   <in-udp-packets-512-bytes>0</in-udp-packets-512-bytes>
   <in-udp-packets-1024-bytes>3540</in-udp-packets-1024-bytes>
   <in-udp-packets-1500-bytes>0</in-udp-packets-1500-bytes>
   <in-udp-packets-3000-bytes>0</in-udp-packets-3000-bytes>
   <in-udp-packets-4500-bytes>0</in-udp-packets-4500-bytes>
   <in-udp-packets-6000-bytes>0</in-udp-packets-6000-bytes>
   <in-udp-packets-9000-bytes>0</in-udp-packets-9000-bytes>
  </global-lan-statistics>
 </Response>
```

brocade-extension/lan-flow-statistics

Description

The LAN per-flow statistics.

Parameters

Table 35: brocade-extension/lan-flow-statistics

Name	Type	Description
flow-index	uint32	flow index associated with the LAN flow. This is a dynamic index associated with the LAN flow. Depending on the LAN flow behavior the index may change and also can get reused after some time but at any given time they will be unique.
slot	uint16	In case of non-chassis system, the slot number is always 0. In case of chassis system, it is the slot number of chassis in which the extension blade is inserted in. In case of chassis, slot number is non-zero value.
dp-id	extension:dp-id-type	Extension Data Path Processor ID associated with flow. Based on platform either it will have a single DP or dual DP. In case of single DP only DP0 is supported, and in case of dual DP both DP0 and DP1 are supported 0 : DP0 1 : DP1.
protocol	extension:l4-protocol-type	Describes that the Layer 4 protocol of the flow.
source-port	inet:port-number	Source port number of the LAN flow.
destination-port	inet:port-number	Remote destination port number of the LAN flow.
source-ip-address	inet:ip-address	Source IP address corresponding to the LAN flow.

Name	Type	Description
destination-ip-address	inet:ip-address	Destination IP address corresponding to the LAN flow.
traffic-control-list-name	extension:extension-name-type	The traffic-control-list name matching the flow filter to allow the traffic.
lan-interface	extension:extension-lan-interface-type	The interface corresponding to the traffic. This could be either a GE port or a LAG name associated with the LAN flow.
vlan-id	extension:vlan-id-type	Specifies the VLAN ID associated with the flow. When not set, this value will show up as 0.
vlan-priority	uint32	Specifies the VLAN priority associated with the flow.
dscp	inet:dscp	DSCP value for the LAN flow.
hcl-flow	boolean	Indicates that a LAN flow is in HCL. In case of UDP flows the value is always set to false as its not applicable. However, in case of TCP flows the value can be either true or false. In case of TCP flows, the value will be seen as true only during the HCL sequence for the duration it's in failovered state and after failback it will be again set to false. true: The flow is in HCL. false: The flow is not in HCL.
active-flow	boolean	Indicates that LAN flow is currently active. true: The flow is active. false: The flow is not active.
start-time	yang:date-and-time	Indicates the LAN flow start time.
end-time	yang:date-and-time	Indicates the LAN flow end time.
rtt	yang:gauge32	TCP round trip time: This leaf is deprecated. Please use new leaf tcp-rtt.
tcp-rtt	yang:gauge64	TCP round trip time.
in-tcp-bytes	yang:zero-based-counter64	Total bytes received.
in-tcp-packets	yang:zero-based-counter64	Total TCP packets received.
out-tcp-bytes	yang:zero-based-counter64	Total bytes sent.
out-tcp-packets	yang:zero-based-counter64	Total TCP packets sent.
tcp-out-of-order-packets	yang:zero-based-counter64	TCP total out of order packets.
tcp-retransmits	yang:zero-based-counter64	TCP retransmits /lost packets.
fast-retransmits	yang:zero-based-counter64	TCP fast retransmits count.
slow-retransmits	yang:zero-based-counter64	TCP slow retransmits count.
duplicate-acknowledgement	yang:zero-based-counter64	TCP duplicate ACK received.
zero-window-count	yang:zero-based-counter64	The count of TCP zero window encountered.
zero-window-maximum-duration	yang:zero-based-counter64	The maximum of zero window duration encountered.
out-bytes-average	yang:gauge64	The throughput in bps for packets sent over the extension tunnel on WAN per 30s average.
in-bytes-average	yang:gauge64	The throughput in bps for packets received via an extension tunnel over WAN per 30s average.

Name	Type	Description
in-drops-lan-session-manager	yang:zero-based-counter64	The number of drops at the ingress from LAN session manager.
out-drops-lan-session-manager	yang:zero-based-counter64	The number of drops at the egress from LAN session manager.
crc-errors	yang:zero-based-counter64	Number of CRC errors encountered.
in-bytes-lan-session-manager	yang:zero-based-counter64	Total bytes received by LAN session manager.
in-packets-lan-session-manager	yang:zero-based-counter64	Total packets received by LAN session manager.
out-bytes-lan-session-manager	yang:zero-based-counter64	Total bytes sent by LAN session manager.
out-packets-lan-session-manager	yang:zero-based-counter64	Total packets sent by LAN session manager.
compression-enabled	boolean	Indicates the LAN flow is running over an extension-tunnel with compression enabled or not. true: The extension-tunnel has compression enabled. false: The extension-tunnel has compression disabled.
in-bytes-lan-compression	yang:zero-based-counter64	Total bytes received by compression engine from LAN.
in-bytes-wan-compression	yang:zero-based-counter64	Total bytes received by compression engine from WAN.
out-bytes-lan-compression	yang:zero-based-counter64	Total bytes sent by compression engine on LAN.
out-bytes-wan-compression	yang:zero-based-counter64	Total bytes sent by compression engine on WAN.
local-host-mss	yang:zero-based-counter32	The local-host-mss is the MSS of the TCP connection at the LAN ingress side connected host. The leaf will be returned in GET response when the value is non-zero.
remote-host-mss	yang:zero-based-counter32	The remote-host-mss is the MSS of the TCP connection at peer extension tunnel endpoint connected host to its the LAN ingress side. The leaf will be returned in GET response when the value is non-zero.
mapped-tunnel	Container	The interface used for extension-tunnel.
- ve-port	extension:ve-interface-type	The VE port of the extension-tunnel interface.
- qos	extension:ip-qos-type	The IP priority QOS associated with the flow.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
 <?xml version="1.0"?>
<Response>
 <lan-flow-statistics>
  <flow-index>7</flow-index>
  <slot>0</slot>
  <dp-id>0</dp-id>
  <protocol>TCP</protocol>
  <source-port>61840</source-port>
  <destination-port>60771</destination-port>
  <source-ip-address>174.10.6.5</source-ip-address>
  <destination-ip-address>165.105.21.5</destination-ip-address>
  <lan-interface>0/6</lan-interface>
 <mapped-tunnel>
  <ve-port>0/12</ve-port>
  <qos>IP-priority-high</qos>
 </mapped-tunnel>
  <vlan-id>0</vlan-id>
  <vlan-priority>0</vlan-priority>
  <dscp>0</dscp>
  <hcl-flow>false</hcl-flow>
  <active-flow>true</active-flow>
  <start-time>2020-01-29T17:56:09-08:00</start-time>
  <rtt>0</rtt>
  <in-tcp-bytes>189636844</in-tcp-bytes>
  <in-tcp-packets>126487</in-tcp-packets>
  <out-tcp-bytes>1176216</out-tcp-bytes>
  <out-tcp-packets>29405</out-tcp-packets>
  <tcp-out-of-order-packets>0</tcp-out-of-order-packets>
  <tcp-retransmits>0</tcp-retransmits>
  <fast-retransmits>0</fast-retransmits>
  <slow-retransmits>0</slow-retransmits>
  <duplicate-acknowledgement>0</duplicate-acknowledgement>
  <zero-window-count>5355</zero-window-count>
  <zero-window-maximum-duration>17</zero-window-maximum-duration>
  <in-bytes-average>0</in-bytes-average>
  <out-bytes-average>0</out-bytes-average>
  <in-drops-lan-session-manager>0</in-drops-lan-session-manager>
  <out-drops-lan-session-manager>0</out-drops-lan-session-manager>
  <in-bytes-lan-session-manager>184577352</in-bytes-lan-session-manager>
  <out-bytes-lan-session-manager>4</out-bytes-lan-session-manager>
  <in-packets-lan-session-manager>0</in-packets-lan-session-manager>
  <out-packets-lan-session-manager>1</out-packets-lan-session-manager>
  <crc-errors>0</crc-errors>
  <local-host-mss>1460</local-host-mss>
  <remote-host-mss>1460</remote-host-mss>
 </lan-flow-statistics>
```

brocade-extension/traffic-control-list

Description

Represents traffic control lists in order to manage IP Extension

LAN flows.

Parameters

Table 36: brocade-extension/traffic-control-list

Name	Type	Description
traffic-control-list-name	extension:extension-name-type	Name of the Traffic-Control-List.
priority	uint32	TCL priority provides an order of precedence to the TCL rule within the overall TCL list. The priority 65535 is a special priority associated or applicable to only the default TCL rule and no other user configured TCL is allowed this value. The priority 0 is a also treated as a special priority and is only to indicate that the priority is not set. A valid user configured TCL priority must use a value from 1 to 65534 only.
admin-state-enabled	boolean	Is TCL admin status enabled. Default: false, Values supported are true/false. false - Disabled true - Enabled Setting the admin-state-enabled value to true will fail if all required parameters are not set. This can include the TCL target information depending on what the action is set to.
cp-dp-synchronized	boolean	Indicates whether an admin-statatus-enabled TCL is synchronized between the CP->DP. In case of error the value will be set to false false - CP-DP synchronizing failed. true - CP-DP is synchronized.
action	extension:tcl-action-type	Set the action for this TCL. The TCL can be programmed to allow a traffic or deny the traffic.
target-ve-port	extension:ve-interface-type	The VE port of the extension-tunnel used for allowing a LAN ingress traffic to be sent over the WAN.
target-qos	extension:ip-qos-type	QoS priority associated with an extension-tunnel to be used to allow a LAN ingress traffic over WAN.
target-slot	uint16	In case of non-chassis system, the slot number is always 0. In case of chassis system, it is the slot number of chassis in which the extension blade is inserted in. In case of chassis, slot number is non-zero value.

Name	Type	Description
target-dp-id	extension:dp-id-type	Extension Data Path Processor ID. Based on platform either it will have a single DP or dual DP. In case of single DP only DP0 is supported, and in case of dual DP both DP0 and DP1 are supported 0 : DP0 1 : DP1.
source-address	union {inet:ip-address;string;}	Source IP address input filter for this TCL.
source-address-prefix-length	brocade-interface-types:ip-prefix-length	The prefix length operator for source IP address input filter.
destination-address	union {inet:ip-address;string;}	Destination IP address input filter for this TCL.
destination-address-prefix-length	brocade-interface-types:ip-prefix-length	The prefix length operator for destination IP address input filter.
port	string	The protocol port input filter for this TCL. The port arguments can be specified as a single port or in case of multiple ports a comma separated list of ports or else a range of ports can be specified or a combination. example : 22 or 600-603 or 300,302,305 The 'application' leaf is mutually exclusive with 'port' leaf and should not be specified together.

Name	Type	Description
application	extension:extension-name-type	The application input filter for this TCL. This includes a list of known apps already present or a user defined app-type name. The 'any' application type name is a special value to identify any application. The 'application' leaf is mutually exclusive with 'port' leaf and should not be specified together. Below are few examples of system defined known application types: CIFS Data-Domain FCIP FTP HTTP HTTPS Isilon-SyncIQ LDAP MS-SQL MySQL NETAPP-SNAP-MIRROR NFS ORACLE-SQL RSYNC SRDF SSH SSL-SHELL TELNET TFTP VERITAS-BACKUP VTS-GRID Control VTS-GRID Data iSCSI
dscp	string	The DSCP input filter for this TCL. The values supported are from 0-63. The value 'any' is default value and is meant to match any dscp value specified.
vlan	string	The VLAN input filter for this TCL. The values supported are from 1-4095. The value 'any' is the default value and is meant to match any vlan-id value specified.
l2cos	string	The L2CoS input filter for this TCL. The valid values are from 0-7. The value 'any' is the default value and is meant to match any value of l2CoS.

Name	Type	Description
l4-protocol	extension:l4-protocol-type	The Layer 4 protocol input filter for this TCL. The value can be a well known protocol string value or otherwise a L4 protocol number. The 'any' protocol string is the default value and is meant to match any L4 protocol value. The valid range for L4 protocol is from 0-255. The list of known protocol string values from system are as below: ICMP ICMP6 TCP UDP VRRP
reset-propagation-enabled	boolean	Is End to End reset propagation for this TCL enabled. Only valid for 'TCP' L4 protocol. Default: false, Values supported are true/false. false - Disabled true - Enabled
segment-preservation-enabled	boolean	Is segment preservation for this TCL enabled. Only valid for 'TCP' L4 protocol. Default: false, Values supported are true/false. false - Disabled true - Enabled
non-terminated-enabled	boolean	Is non terminated traffic for this TCL enabled. Only valid for 'TCP' L4 protocol. Default: false, Values supported are true/false. false - Disabled true - Enabled
hit-count	yang:zero-based-counter64	Total number of times this TCL rule was hit.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
```

```
  <traffic-control-list>
   <traffic-control-list-name>Test2</traffic-control-list-name>
   <priority>2</priority>
   <admin-state-enabled>true</admin-state-enabled>
   <cp-dp-synchronized>true</cp-dp-synchronized>
   <target-ve-port>8/17</target-ve-port>
   <target-qos>IP-priority-medium</target-qos>
   <vlan>any</vlan>
   <l2cos>any</l2cos>
   <dscp>any</dscp>
   <source-address>any</source-address>
   <destination-address>any</destination-address>
   <l4-protocol>any</l4-protocol>
   <port>any</port>
   <application>any</application>
   <action>allow</action>
   <reset-propagation-enabled>false</reset-propagation-enabled>
   <segment-preservation-enabled>false</segment-preservation-enabled>
   <non-terminated-enabled>false</non-terminated-enabled>
   <hit-count>13297</hit-count>
  </traffic-control-list>
  <traffic-control-list>
   <traffic-control-list-name>default</traffic-control-list-name>
   <priority>65535</priority>
   <admin-state-enabled>true</admin-state-enabled>
   <cp-dp-synchronized>true</cp-dp-synchronized>
   <vlan>any</vlan>
   <l2cos>any</l2cos>
   <dscp>any</dscp>
   <source-address>any</source-address>
   <destination-address>any</destination-address>
   <l4-protocol>any</l4-protocol>
   <port>any</port>
   <application>any</application>
   <action>deny</action>
   <reset-propagation-enabled>false</reset-propagation-enabled>
   <segment-preservation-enabled>false</segment-preservation-enabled>
   <non-terminated-enabled>false</non-terminated-enabled>
   <hit-count>23383</hit-count>
  </traffic-control-list>
 </Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-fabric-traffic-controller/fabric-traffic-controller-ag-f-port

Description

A list of AG F_Ports entries relevant to Fabric Traffic Controller.

Parameters

Table 37: brocade-fabric-traffic-controller/fabric-traffic-controller-ag-f-port

Name	Type	Description
edc-owner	fibrechannel:fcid-hex-string-type	N_Port ID of the device that owns the EDC (diagnostic capabilities) information that applies to this N_Port. The value may or may not match the value of the associated 'port-id' leaf of a given list entry. If the N_Port has not initiated an EDC ELS exchange, then the value returned is '0x000000'.
n-port-diagnostic-capabilities	Container	The Diagnostic Capabilities of the N_Port (device).
tx-signal-capability	congestion-detection-signal-type	The transmit congestion detection signal capabilities of the port.
tx-signal-cycle	congestion-detection-signal-cycles-type	The transmit congestion detection cycle applicable to the port.
tx-signal-scale	congestion-detection-signal-scale-type	The transmit congestion detection scale applicable to the port.
rx-signal-capability	congestion-detection-signal-type	The receive congestion detection signal capabilities of the port.
rx-signal-cycle	congestion-detection-signal-cycles-type	The receive congestion detection cycle applicable to the port.
rx-signal-scale	congestion-detection-signal-scale-type	The receive congestion detection scale applicable to the port.
degrade-activate-threshold	uint32	The value of a 32-bit register that specifies a symbol error count. The value here controls the threshold used to activate the Remote Degrade Bit being sent in the Alignment Marker field.
degrade-deactivate-threshold	uint32	The value of a 32-bit register that specifies a symbol error count. The value here controls the threshold used to deactivate the Remote Degrade Bit being sent in the Alignment Marker field.
fec-degrade-interval	uint32	The value of a 32-bit register that specifies the number of RS-FEC code words that make up a Degrade Interval.
f-port-diagnostic-capabilities	Container	The Diagnostic Capabilities of the F_Port (switch).
tx-signal-capability	congestion-detection-signal-type	The transmit congestion detection signal capabilities of the port.
tx-signal-cycle	congestion-detection-signal-cycles-type	The transmit congestion detection cycle applicable to the port.

Name	Type	Description
tx-signal-scale	congestion-detection-signal-scale-type	The transmit congestion detection scale applicable to the port.
rx-signal-capability	congestion-detection-signal-type	The receive congestion detection signal capabilities of the port.
rx-signal-cycle	congestion-detection-signal-cycles-type	The receive congestion detection cycle applicable to the port.
rx-signal-scale	congestion-detection-signal-scale-type	The receive congestion detection scale applicable to the port.
degrade-activate-threshold	uint32	The value of a 32-bit register that specifies a symbol error count. The value here controls the threshold used to activate the Remote Degrade Bit being sent in the Alignment Marker field.
degrade-deactivate-threshold	uint32	The value of a 32-bit register that specifies a symbol error count. The value here controls the threshold used to deactivate the Remote Degrade Bit being sent in the Alignment Marker field.
fec-degrade-interval	uint32	The value of a 32-bit register that specifies the number of RS-FEC code words that make up a Degrade Interval.
applied-signaling-capabilities	Container	The congestion detection signal capabilities that are common between the N_Port (device) and F_Port (switch). The values are presented from the perspective of the switch, and represent the set of values that are actually programmed in hardware.
tx-signal-capability	congestion-detection-signal-type	The transmit congestion detection signal capabilities of the port.
tx-signal-cycle	congestion-detection-signal-cycles-type	The transmit congestion detection cycle applicable to the port.
tx-signal-scale	congestion-detection-signal-scale-type	The transmit congestion detection scale applicable to the port.
rx-signal-capability	congestion-detection-signal-type	The receive congestion detection signal capabilities of the port.
rx-signal-cycle	congestion-detection-signal-cycles-type	The receive congestion detection cycle applicable to the port.
rx-signal-scale	congestion-detection-signal-scale-type	The receive congestion detection scale applicable to the port.
port-index	fibrechannel:user-port-number-type	The port index of the AG F_Port.
ag-mapped-n-port-index	fibrechannel:user-port-number-type	The port index of the AG N_Port mapped to this AG F_Port.
congestion-states	Container	A list of congestion symptoms, or 'states', exhibited on the port.
- congestion-state	congestion-state-type	A congestion state entry.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fabric-traffic-controller-ag-f-port>
        <port-index>1</port-index>
        <ag-mapped-n-port-index>20</ag-mapped-n-port-index>
        <congestion-states>
            <congestion-state>none</congestion-state>
        </congestion-states>
        <edc-owner>0x000000</edc-owner>
    </fabric-traffic-controller-ag-f-port>
 ...
    <fabric-traffic-controller-ag-f-port>
        <port-index>13</port-index>
        <ag-mapped-n-port-index>22</ag-mapped-n-port-index>
        <congestion-states>
            <congestion-state>none</congestion-state>
        </congestion-states>
        <edc-owner>0x000000</edc-owner>
    </fabric-traffic-controller-ag-f-port>
</Response>
```

brocade-fabric-traffic-controller/fabric-traffic-controller-ag-n-port

Description

A list of AG N_Ports entries relevant to Fabric Traffic Controller.

Parameters

Table 38: brocade-fabric-traffic-controller/fabric-traffic-controller-ag-n-port

Name	Type	Description
port-index	fibrechannel:user-port-number-type	User Port Index of the AG N_Port.
port-id	fibrechannel:fcid-hex-string-type	N_Port ID (Fibre Channel Address) of the AG N_Port.
ag-n-port-notification-capabilities	Container	A list of capabilities that are relevant to Fabric Notifications and supported by the Access Gateway.
- capability	notification-capabilities-type	A Fabric Notification Capability entry.

Name	Type	Description
fabric-neighbor-notification-capabilities	Container	A list of capabilities that are relevant to Fabric Notifications and supported by the neighbor switch in the Fabric.
- capability	notification-capabilities-type	A Fabric Notification Capability entry.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fabric-traffic-controller-ag-n-port>
        <port-index>18</port-index>
        <port-id>0xec0d00</port-id>
        <ag-n-port-notification-capabilities>
            <capability>agpin</capability>
        </ag-n-port-notification-capabilities>
        <fabric-neighbor-notification-capabilities>
            <capability>agpin</capability>
        </fabric-neighbor-notification-capabilities>
    </fabric-traffic-controller-ag-n-port>
...
    <fabric-traffic-controller-ag-n-port>
        <port-index>23</port-index>
        <port-id>0xed0400</port-id>
        <ag-n-port-notification-capabilities>
            <capability>agpin</capability>
        </ag-n-port-notification-capabilities>
        <fabric-neighbor-notification-capabilities>
            <capability>agpin</capability>
        </fabric-neighbor-notification-capabilities>
    </fabric-traffic-controller-ag-n-port>
</Response>
```

brocade-fabric-traffic-controller/fabric-traffic-controller-device

Description

A list of device entries applicable to Fabric Traffic Controller.

Parameters

Table 39: brocade-fabric-traffic-controller/fabric-traffic-controller-device

Name	Type	Description
edc-owner	fibrechannel:fcid-hex-string-type	N_Port ID of the device that owns the EDC (diagnostic capabilities) information that applies to this N_Port. The value may or may not match the value of the associated 'port-id' leaf of a given list entry. If the N_Port has not initiated an EDC ELS exchange, then the value returned is '0x000000'.
n-port-diagnostic-capabilities	Container	The Diagnostic Capabilities of the N_Port (device).
tx-signal-capability	congestion-detection-signal-type	The transmit congestion detection signal capabilities of the port.
tx-signal-cycle	congestion-detection-signal-cycles-type	The transmit congestion detection cycle applicable to the port.
tx-signal-scale	congestion-detection-signal-scale-type	The transmit congestion detection scale applicable to the port.
rx-signal-capability	congestion-detection-signal-type	The receive congestion detection signal capabilities of the port.
rx-signal-cycle	congestion-detection-signal-cycles-type	The receive congestion detection cycle applicable to the port.
rx-signal-scale	congestion-detection-signal-scale-type	The receive congestion detection scale applicable to the port.
degrade-activate-threshold	uint32	The value of a 32-bit register that specifies a symbol error count. The value here controls the threshold used to activate the Remote Degrade Bit being sent in the Alignment Marker field.
degrade-deactivate-threshold	uint32	The value of a 32-bit register that specifies a symbol error count. The value here controls the threshold used to deactivate the Remote Degrade Bit being sent in the Alignment Marker field.
fec-degrade-interval	uint32	The value of a 32-bit register that specifies the number of RS-FEC code words that make up a Degrade Interval.
f-port-diagnostic-capabilities	Container	The Diagnostic Capabilities of the F_Port (switch).
tx-signal-capability	congestion-detection-signal-type	The transmit congestion detection signal capabilities of the port.
tx-signal-cycle	congestion-detection-signal-cycles-type	The transmit congestion detection cycle applicable to the port.
tx-signal-scale	congestion-detection-signal-scale-type	The transmit congestion detection scale applicable to the port.
rx-signal-capability	congestion-detection-signal-type	The receive congestion detection signal capabilities of the port.
rx-signal-cycle	congestion-detection-signal-cycles-type	The receive congestion detection cycle applicable to the port.

Name	Type	Description
rx-signal-scale	congestion-detection-signal-scale-type	The receive congestion detection scale applicable to the port.
degrade-activate-threshold	uint32	The value of a 32-bit register that specifies a symbol error count. The value here controls the threshold used to activate the Remote Degrade Bit being sent in the Alignment Marker field.
degrade-deactivate-threshold	uint32	The value of a 32-bit register that specifies a symbol error count. The value here controls the threshold used to deactivate the Remote Degrade Bit being sent in the Alignment Marker field.
fec-degrade-interval	uint32	The value of a 32-bit register that specifies the number of RS-FEC code words that make up a Degrade Interval.
applied-signaling-capabilities	Container	The congestion detection signal capabilities that are common between the N_Port (device) and F_Port (switch). The values are presented from the perspective of the switch, and represent the set of values that are actually programmed in hardware.
tx-signal-capability	congestion-detection-signal-type	The transmit congestion detection signal capabilities of the port.
tx-signal-cycle	congestion-detection-signal-cycles-type	The transmit congestion detection cycle applicable to the port.
tx-signal-scale	congestion-detection-signal-scale-type	The transmit congestion detection scale applicable to the port.
rx-signal-capability	congestion-detection-signal-type	The receive congestion detection signal capabilities of the port.
rx-signal-cycle	congestion-detection-signal-cycles-type	The receive congestion detection cycle applicable to the port.
rx-signal-scale	congestion-detection-signal-scale-type	The receive congestion detection scale applicable to the port.
f-port-wwn	fibrechannel:wwn-type	F_Port Worldwide Name of the F_Port to which the N_Port is attached.
port-index	fibrechannel:user-port-number-type	User Port Index.
rdf-registrations	Container	A list of ELS-to-ELS-descriptor pairs that have been registered via the RDF ELS exchange.
- rdf-registration	els-descriptor-type	A list entry for an RDF-registered ELS-to-ELS-descriptor pair.
congestion-states	Container	A list of congestion symptoms, or 'states', exhibited on the port.
- congestion-state	congestion-state-type	A congestion state entry.
fpin-send-statistics	Container	Quantitative data pertaining to FPIN ELS exchanges originated by the switch.
peer-congestion-count	yang:counter32	Peer-Congestion descriptor count.
congestion-count	yang:counter32	Congestion descriptor count.
link-integrity-count	yang:counter32	Link Integrity descriptor count.
delivery-count	yang:counter32	Delivery descriptor count.
fpin-receive-statistics	Container	Quantitative data pertaining to FPIN ELS exchanges received by the switch.
peer-congestion-count	yang:counter32	Peer-Congestion descriptor count.

Name	Type	Description
congestion-count	yang:counter32	Congestion descriptor count.
link-integrity-count	yang:counter32	Link Integrity descriptor count.
delivery-count	yang:counter32	Delivery descriptor count.
n-port-id	fibrechannel:fcid-hex-string-type	N_Port ID (Fibre Channel Address) of the device.
n-port-wwn	fibrechannel:wwn-type	N_Port Worldwide Name of the device.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fabric-traffic-controller-device>
        <n-port-id>0x2b9200</n-port-id>
        <n-port-wwn>10:00:00:10:9b:8f:2b:c7</n-port-wwn>
        <f-port-wwn>20:92:c4:f5:7c:4a:ac:6c</f-port-wwn>
        <port-index>146</port-index>
        <rdf-registrations>
            <rdf-registration>fpin-link-integrity</rdf-registration>
        </rdf-registrations>
        <congestion-states>
            <congestion-state>none</congestion-state>
        </congestion-states>
        <edc-owner>0x000000</edc-owner>
        <fpin-send-statistics>
            <peer-congestion-count>0</peer-congestion-count>
            <congestion-count>0</congestion-count>
            <link-integrity-count>0</link-integrity-count>
            <delivery-count>0</delivery-count>
        </fpin-send-statistics>
        <fpin-receive-statistics>
            <peer-congestion-count>0</peer-congestion-count>
            <congestion-count>0</congestion-count>
            <link-integrity-count>0</link-integrity-count>
            <delivery-count>0</delivery-count>
        </fpin-receive-statistics>
    </fabric-traffic-controller-device>
...
```

```
    <fabric-traffic-controller-device>
        <n-port-id>0x2bad00</n-port-id>
        <n-port-wwn>10:00:00:10:9b:8f:2b:c6</n-port-wwn>
        <f-port-wwn>20:ad:c4:f5:7c:4a:ac:6c</f-port-wwn>
        <port-index>173</port-index>
        <rdf-registrations>
            <rdf-registration>fpin-link-integrity</rdf-registration>
        </rdf-registrations>
        <congestion-states>
            <congestion-state>none</congestion-state>
        </congestion-states>
        <edc-owner>0x000000</edc-owner>
        <fpin-send-statistics>
            <peer-congestion-count>0</peer-congestion-count>
            <congestion-count>0</congestion-count>
            <link-integrity-count>0</link-integrity-count>
            <delivery-count>0</delivery-count>
        </fpin-send-statistics>
        <fpin-receive-statistics>
            <peer-congestion-count>0</peer-congestion-count>
            <congestion-count>0</congestion-count>
            <link-integrity-count>0</link-integrity-count>
            <delivery-count>0</delivery-count>
        </fpin-receive-statistics>
    </fabric-traffic-controller-device>
</Response>
```

brocade-fabric/access-gateway

Description

Lists all the access gateway connect to the fabric.

This can be retrieved by querying on any of switches part of the Fabric.

Parameters

Table 40: brocade-fabric/access-gateway

Name	Type	Description
switch-wwn	fibrechannel:wwn-type	The world wide name (WWN) of the connected Access Gateway.
user-friendly-name	string	The ASCII name assigned to the Access Gateway by the administrator. 1 to 30 alphanumeric characters plus hyphens, periods, and underscores. Spaces are not allowed. A switch name can begin with a letter or number, but a switch name that begins with a numeric character (0-9) must also have at least an underscore (_), hyphen (-), period (.), or alphabetic character (A-Z, a-z). A switch name with only numeric characters is not valid.
firmware-version	string	A human readable string identifying the firmware version running on the access gateway switch.

Name	Type	Description
is-edge-ag	boolean	Whether the Connected Access Gateway switch is Core AG or Edge AG. true : device is an edge AG device false : device is not an edge AG device
ip-addresses	Container	List of out-of-band IP addresses assigned to the Ethernet port on the Access Gateway.
- ip-address	union {inet:ip-address;inet:ip-prefix;}	List of out-of-band IP addresses assigned to the Ethernet port of the switch.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <access-gateway>
        <switch-wwn>10:00:c1:f1:1c:d1:be:ae</switch-wwn>
        <user-friendly-name>G610</user-friendly-name>
        <ip-addresses>
            <ip-address>10.10.10.11</ip-address>
            <ip-address>2220:102:2:fa02:c2f2:2cff:fed2:beae</ip-address>
        </ip-addresses>
        <firmware-version>v9.1.0</firmware-version>
        <is-edge-ag>false</is-edge-ag>
    </access-gateway>
    <access-gateway>
        <switch-wwn>10:00:d3:1f:cc:3a:3d:e3</switch-wwn>
        <user-friendly-name>G720</user-friendly-name>
        <ip-addresses>
            <ip-address>10.10.10.13</ip-address>
            <ip-address>2330:100:3:fa03:da3f:ccff:fe3a:3de3</ip-address>
        </ip-addresses>
        <firmware-version>v9.1.0</firmware-version>
        <is-edge-ag>false</is-edge-ag>
    </access-gateway>
</Response>
```

brocade-fabric/fabric-switch

Description

The list of configured switches in the fabric.

Parameters

Table 41: brocade-fabric/fabric-switch

Name	Type	Description
name	fibrechannel:wwn-type	The Fibre Channel WWN of the switch.
switch-user-friendly-name	string	An ASCII name assigned to the switch by the administrator.
chassis-wwn	fibrechannel:wwn-type	The chassis WWN.
chassis-user-friendly-name	union {fibrechannel:chassis-user-friendly-name-type;fibrechannel:brocade-empty-string-type;}	An ASCII name assigned to the switch chassis by the administrator.
domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
principal	uint8	Indicates if this switch is the fabric principal switch. The possible values are: 0 : Switch is not principal switch 1 : Switch is principal switch
is-principal	boolean	Indicates if this switch is the fabric principal switch. true : Switch is the principal switch in the fabric. false : Switch is not the principal switch in the fabric.
fcid	fibrechannel:fcid-type	The destination ID (D_ID) of the switch (decimal format). Deprecated: Please use 'fcid-hex' leaf instead.
fcid-hex	fibrechannel:fcid-hex-string-type	The destination ID (D_ID) of the switch (hexadecimal format).
ip-address	inet:ipv4-address	IPV4 address for the switch
fcip-address	inet:ipv4-address	The IP address the switch is using for Fibre Channel over IP.
ipv6-address	inet:ipv6-address	IPV6 address for the switch
firmware-version	string	A human readable string identifying the firmware version running on the switch.
path-count	yang:gauge32	The number of paths available to each remote domain. Deprecated: Please use path-count leaf from topology-domain list.
switch-model	string	The model name of the switch in the fabric.
serial-number	fru:serial-number-type	A printable ASCII string that specifies the factory serial number of the chassis.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <fabric-switch>
  <name>10:10:14:15:1c:9e:3b:c8</name>
  <chassis-wwn>10:10:14:15:1c:9e:3c:07</chassis-wwn>
  <domain-id>1</domain-id>
  <fcid>16776199</fcid>
  <switch-user-friendly-name>G610_81</switch-user-friendly-name>
  <chassis-user-friendly-name>BrocadeG610</chassis-user-friendly-name>
  <firmware-version>v820</firmware-version>
  <ip-address>10.10.10.1</ip-address>
  <fcip-address>0.0.0.0</fcip-address>
  <ipv6-address>::</ipv6-address>
  <principal>1</principal>
 </fabric-switch>
 <fabric-switch>
  <name>10:10:14:15:1c:a2:1f:40</name>
  <chassis-wwn>10:10:14:15:1c:a2:1f:7f</chassis-wwn>
  <domain-id>3</domain-id>
  <fcid>16776195</fcid>
  <switch-user-friendly-name>G610_82</switch-user-friendly-name>
  <chassis-user-friendly-name>BrocadeG610</chassis-user-friendly-name>
  <firmware-version>v820</firmware-version>
  <ip-address>10.10.10.2</ip-address>
  <fcip-address>0.0.0.0</fcip-address>
  <ipv6-address>::</ipv6-address>
  <principal>0</principal>
 </fabric-switch>
</Response>
```

brocade-fdmi/hba

Description

List of HBA attributes registered with FDMI.

Parameters

Table 42: brocade-fdmi/hba

Name	Type	Description
hba-id	fibrechannel:wwn-type	The hba-id resource is a 64-bit Name_Identifier that is uniquely associated with the HBA among all HBAs in the same Fibre Channel interaction space (see FC-FS). The HBA Identifier for an HBA may be the same as the Name_Identifier of an Nx_Port on the HBA if the required persistence is satisfied. Once an HBA has registered a Name_Identifier as its HBA Identifier, that Name_Identifier persists (e.g., across power cycles) as the HBA Identifier for the HBA.
domain-id	fibrechannel:domain-id-type	The domain-id resource is the domain the HBA is directly attached to.
manufacturer	string	The manufacturer resource contains a Printable ASCII String that specifies the manufacturer of the host adapter. The value may match the name by which the manufacturer identifies itself in a telephone directory. The registration of this attribute is optional.
serial-number	string	The serial-number resource contains a Printable ASCII String that specifies the serial number of the host adapter. The value should match a serial number engraved or printed on the host bus adapter, if there is any. The registration of this attribute is optional.
model	string	The model resource contains a Printable ASCII String that specifies the model of the host adapter. The value may match an encoded string used on purchase orders to identify the host adapter model. Some management applications limit this resource to 63 bytes. The registration of this attribute is optional.
model-description	string	The Model Description resource contains a Printable ASCII String that describes the model of the host adapter. The value may provide more detailed or human oriented identification of the model of the host bus adapter than the model resource does. The registration of this attribute is optional.
node-name	fibrechannel:wwn-type	The node-name resource contains an 8 byte value that identifies the node that contains the Nx_Ports on the host adapter. If all Nx_Ports on the host bus adapter have the same node name, the node-name resource returned for the host bus adapter shall match the node-name resource for its Nx_Ports. If not all Nx_Ports on the host bus adapter have the same node name, the node-name resource will not be returned.
node-symbolic-name	string	The node-symbolic-name resource is the name registered with the Name Server as a Name Server Symbolic Node Name object and is subject to all the description and constraints of that object. If not all Nx_Ports on the host bus adapter have the same node name, the node-symbolic-name resource will not be returned. The registration of this attribute is optional.

Name	Type	Description
hardware-version	string	The hardware-version resource contains a Printable ASCII String that identifies the hardware version level of the host adapter. Some management applications limit this to a length of 63 characters. The registration of this attribute is optional.
driver-version	string	The driver-version resource contains a Printable ASCII String that identifies the version level of the driver software controlling a host adapter. If a host bus adapter is concurrently under control of multiple driver software modules with different versions, this resource may indicate the versions for more than one driver module. The registration of this attribute is optional.
option-rom-version	string	The option-rom-version resource contains a Printable ASCII String that identifies the option ROM or BIOS version of a host adapter. The registration of this attribute is optional.
firmware-version	string	The firmware-version resource contains a Printable ASCII String that identifies the version of firmware executed by a host adapter. If a host bus adapter contains and has the capability to execute multiple firmware modules with different versions, this resource may indicate the versions for more than one firmware module. The registration of this attribute is optional.
os-name-and-version	string	The os-name-and-version resource contains a Printable ASCII String that describes the type and version of the operating system controlling the host bus adapter. The registration of this attribute is optional.
max-ct-payload	uint32	The max-ct-payload resource is 32-bit unsigned integer equal to the maximum size CT payload in 32-bit words, including all CT headers but no FC frame header(s), that may be sent or received by application software resident in the host in which the host bus adapter is installed. If the host bus adapter does not support generic CT capability for application software on the host in which it is installed, this resource will not be returned. The registration of this attribute is optional.
vendor-id	string	The vendor-id resource contains the T10 Vendor ID of the manufacturer of the HBA, or an OEM of the HBA. The registration of this attribute is optional.
vendor-specific-info	fibrechannel:brocade-hex-string-type	The vendor-specific-info resource contains a value with vendor specific use. The registration of this attribute is optional.
number-of-ports	uint32	The number-of-ports resource contains a value equal to the number of Nx_Ports on the HBA. The registration of this attribute is optional.
fabric-name	fibrechannel:wwn-type	The fabric-name resource contains an 8 byte binary value equal to the Fabric_Name of the Fabric associated with the HBA. If the HBA is associated with more than one Fabric, this resource will not be provided. The registration of this attribute is optional.
boot-bios-version	string	The boot-bios-version resource contains a Printable ASCII String that describes the identification and version of a boot BIOS provided by the HBA. The registration of this attribute is optional.

Name	Type	Description
boot-bios-enabled	uint8	Deprecated: Please reference the boot-bios-enabled-v2 leaf instead. The boot-bios-enabled resource indicates whether a boot BIOS provided by the HBA is enabled or disabled. The registration of this attribute is optional. 0 : Disabled 1 : Enabled
boot-bios-enabled-v2	boolean	The boot-bios-enabled-v2 resource indicates whether a boot BIOS provided by the HBA is enabled or disabled. The registration of this attribute is optional. false : Disabled true : Enabled
hba-port-list	Container	A list of port WWNs associated with the HBA. Corresponds with the port-name resource under the port list.
- wwn	fibrechannel:wwn-type	The WWN of the port on the HBA. Corresponds to the port-name resource under the port list.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.


```
{
    "Response": {
        "hba": [
            {
                "hba-id": "21:00:00:0e:1e:07:03:48",
                "domain-id": 43,
                "manufacturer": "CAVIUM, INC.",
                "serial-number": "RFE1248G93391",
                "model": "QLE2670",
                "model-description": "QLE2670 QLogic 1-port 16Gb Fibre Channel Adapter",
                "node-name": "20:00:00:0e:1e:07:03:48",
                "node-symbolic-name": "QLE2670 FW:v6.02.00 DVR:v9.2.9.23",
                "hardware-version": "HD8310405-01  F",
                "driver-version": "9.2.9.23",
                "option-rom-version": "5.22",
                "firmware-version": "6.02.00",
                "os-name-and-version": "Windows Server 2016 Standard for Intel64",
                "max-ct-payload": 2048,
                "vendor-id": "QLOGIC",
```

```
                "number-of-ports": 1,
                "fabric-name": "10:00:c4:f5:7c:4a:ac:6c",
                "boot-bios-version": "3.14",
                "hba-port-list": {
                    "wwn": [
                        "21:00:00:0e:1e:07:03:48"
                    ]
                }
            },
...
            {
                "hba-id": "10:00:00:10:9b:8f:2d:ea",
                "domain-id": 237,
                "manufacturer": "Emulex Corporation",
                "serial-number": "FP92953378",
                "model": "LPe35002-M2",
                "model-description": "Emulex LPe35002-M2 2-Port 32Gb Fibre Channel Adapter",
                "node-name": "20:00:00:10:9b:8f:2d:ea",
                "node-symbolic-name": "Emulex LPe35002-M2 FV12.8.351.6901 DV12.6.165.0 HN:W16-68166 OS:Windows
 2016",
                "hardware-version": "0000000D",
                "driver-version": "12.6.165.0",
                "option-rom-version": "12.8.351.6901",
                "firmware-version": "12.8.351.6901",
                "os-name-and-version": "Windows 2016",
                "max-ct-payload": 524288,
                "vendor-id": "Emulex",
                "number-of-ports": 1,
                "fabric-name": "10:00:88:94:71:42:7b:d5",
                "boot-bios-version": "12.8.351.6901",
                "hba-port-list": {
                    "wwn": [
                        "10:00:00:10:9b:8f:2d:ea"
                    ]
                }
            }
        ]
    }
}
```


brocade-fdmi/port

Description

List of HBA port attributes registered with FDMI.

Parameters

Table 43: brocade-fdmi/port

Name	Type	Description
port-name	fibrechannel:wwn-type	The port-name resource contains a WWN that identifies the port associated with the host bus adapter specified by the hba-id resource.
hba-id	fibrechannel:wwn-type	The hba-id resource is a 64-bit Name_Identifier that is uniquely associated with the HBA among all HBAs in the same Fibre Channel interaction space (see FC-FS). The HBA Identifier for an HBA may be the same as the Name_Identifier of an Nx_Port on the HBA if the required persistence is satisfied. Once an HBA has registered a Name_Identifier as its HBA Identifier, that Name_Identifier persists (e.g., across power cycles) as the HBA Identifier for the HBA. Corresponds to the hba-id resource under the hba list.
domain-id	fibrechannel:domain-id-type	The domain-id resource is the domain the HBA port is directly attached to.
port-symbolic-name	string	The port-symbolic-name resource is the Symbolic Port Name object registered with the Name Server and is subject to all the description and constraints of that object as defined in FC-GS. The registration of this attribute is optional.
port-id	fibrechannel:fcid-hex-string-type	The port-id resource is the Port Identifier object registered with the Name Server and is subject to all the description and constraints of that object as defined in FC-GS. The registration of this attribute is optional.
port-type	fibrechannel:port-type-string-type	The port-type resource is the Port Type object registered with the Name Server and is subject to all the description and constraints of that object as defined in FC-GS. The registration of this attribute is optional.
supported-class-of-service	fibrechannel:class-of-service-type	The supported-class-of-service resource is the Class of Service object registered with the Name Server and is subject to all the description and constraints of that object as defined in FC-GS. The registration of this attribute is optional.
supported-fc4-type	fibrechannel:fc4-type-type	The supported-fc4-type resource indicates support for any FC-4 type that the Nx_Port is able to be configured to support. The registration of this attribute is optional.
active-fc4-type	fibrechannel:fc4-type-type	The active-fc4-type resource is the FC-4 Types object registered with the Name Server and is subject to all the description and constraints of that object as defined in FC-GS. The resource indicates support for any FC-4 type that the Nx_Port is currently configured to support. The registration of this attribute is optional.
supported-speed	fibrechannel:speed-type	Deprecated: Please reference the supported-protocol-speeds leaf-list instead. The supported-speed resource indicates the supported transmission speeds of the Nx_Port. The registration of this attribute is optional.

Name	Type	Description
current-port-speed	fibrechannel:speed-type	Deprecated: Please reference the current-protocol-speed leaf instead. The current-port-speed resource indicates the current transmission speed of the Nx_Port. If the HBA Management Server is able to determine a speed, the value returned for the current-port-speed resource will be the speed determined by the HBA Management Server, regardless of any value registered. The registration of this attribute is optional.
current-protocol-speed	fibrechannel:protocol-speed-type	The current-protocol-speed resource indicates the current transmission speed of the Nx_Port. If the HBA Management Server is able to determine a speed, the value returned for the current-protocol-speed resource will be the speed determined by the HBA Management Server, regardless of any value registered. The registration of this attribute is optional.
maximum-frame-size	uint32	The maximum-frame-size resource is a 32-bit unsigned integer the value of which is the maximum FC frame payload in bytes. This does not include the FC header but includes any optional headers. The registration of this attribute is optional.
os-device-name	string	The os-device-name resource contains a Printable ASCII String that is recognized as a reference to the Nx_Port by the OS that controls it. If there are several such OS device names that reference the same Nx_Port, this resource may be a comma-separated list of as many such names as fit in 255 bytes. The registration of this attribute is optional.
host-name	string	The host-name resource contains a Printable ASCII String that describes the name of the host associated with the Nx_Port. If there are several such names that reference the same host, this resource may be a comma-separated list of as many such names as fit in 255 bytes. The registration of this attribute is optional.
node-name	fibrechannel:wwn-type	The node-name resource contains an 8 byte value that identifies the node that contains the Nx_Port. The node_name resource is equal to the Node_Name registered with the Name Server for the Nx_Port.
fabric-name	fibrechannel:wwn-type	The fabric-name resource contains an 8 byte binary value equal to the Fabric_Name of the Fabric associated with the Nx_Port (see FC-SW). The registration of this attribute is optional.
port-state	fibrechannel:brocade-hex-string-type	The port-state resource is the the Port State (see SM-HBA) for the specified Nx_Port. The port-state is an integer where the value indicates the current state of the Nx_Port. The sequence and timing of port-states that are exhibited due to errors or transient conditions is vendor specific. The registration of this attribute is optional. 0x0 : Undefined 0x1 : Unknown 0x2 : Fully Operational 0x3 : Administratively Offline 0x4 : Bypassed 0x5 : In Diagnostics Mode 0x6 : Link Down 0x7 : Phy Error 0x8 : Loopback 0x9 : Degraded, but Operational Mode

Name	Type	Description
number-of-discovered-ports	uint32	The number-of-discovered-ports resource is the number of FC_Ports that are visible to the Nx_Port identified in the request. At a minimum, this is the number of FC_Ports mapped to a device, but may not reflect all of the nodes on a network. The registration of this attribute is optional.
vsa-service-category	string	Vendor-Specific service category resource. The registration of this attribute is optional.
vsa-guid	string	Vendor-Specific guid resource. The registration of this attribute is optional.
vsa-version	string	Vendor-Specific version resource. The registration of this attribute is optional.
vsa-product-name	string	Vendor-Specific product name resource. The registration of this attribute is optional.
vsa-port-info	string	Vendor-Specific port info resource. The registration of this attribute is optional.
vsa-qos-supported	string	Vendor-Specific qos supported resource. The registration of this attribute is optional.
vsa-security	string	Vendor-Specific security resource. The registration of this attribute is optional.
vsa-storage-array-family	string	Vendor-Specific storage array family resource. The registration of this attribute is optional.
vsa-storage-array-name	string	Vendor-Specific storage array name resource. The registration of this attribute is optional.
vsa-storage-array-system-model	string	Vendor-Specific storage array system model resource. The registration of this attribute is optional.
vsa-storage-array-os	string	Vendor-Specific storage array os resource. The registration of this attribute is optional.
vsa-storage-array-node-count	uint32	Vendor-Specific storage array node count resource. The registration of this attribute is optional.
vsa-end-to-end-version	string	Vendor-Specific end-to-end management version. The registration of this attribute is optional.
supported-protocol-speeds	Container	A list of the supported network speeds.
- supported-protocol-speed	fibrechannel:protocol-speed-type	The supported-protocol-speed resource indicates the supported transmission speeds of the Nx_Port. The registration of this attribute is optional.
vsa-storage-array-nodes	Container	A list of storage array nodes.
- nodes	string	Vendor-Specific storage array node resource. The registration of this attribute is optional.
vsa-connected-ports	Container	A list of connected ports.
- wwns	fibrechannel:wwn-type	Vendor-Specific connected port resource. The registration of this attribute is optional.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <port>
  <port-name>21:00:00:0e:1e:07:03:48</port-name>
  <hba-id>21:00:00:0e:1e:07:03:48</hba-id>
  <domain-id>43</domain-id>
  <port-symbolic-name>QLE2670 FW:v6.02.00 DVR:v9.2.9.23</port-symbolic-name>
  <port-id>0x2baa00</port-id>
  <port-type>n-port</port-type>
  <supported-class-of-service>class-3</supported-class-of-service>
  <supported-fc4-type>FCP</supported-fc4-type>
  <active-fc4-type>FCP</active-fc4-type>
  <supported-speed>speed-4-gfc speed-8-gfc speed-16-gfc</supported-speed>
  <current-port-speed>speed-16-gfc</current-port-speed>
  <maximum-frame-size>2048</maximum-frame-size>
  <os-device-name>QLogic Adapter</os-device-name>
  <host-name>W16-BRM68164</host-name>
  <node-name>20:00:00:0e:1e:07:01:41</node-name>
  <fabric-name>10:00:c1:f5:1c:1a:ac:1c</fabric-name>
  <port-state>0x2</port-state>
  <number-of-discovered-ports>0</number-of-discovered-ports>
 </port>
...
 <port>
  <port-name>10:00:00:10:1b:1f:1b:c1</port-name>
  <hba-id>10:00:00:10:1b:1f:1b:c1</hba-id>
  <domain-id>43</domain-id>
  <port-symbolic-name>Emulex PPN-10:00:00:10:1B:1F:1B:C1</port-symbolic-name>
  <port-id>0x2bad00</port-id>
  <port-type>n-port</port-type>
  <supported-class-of-service>class-3</supported-class-of-service>
  <supported-fc4-type>0x01 FCP FC-CT FC-NVMe</supported-fc4-type>
  <active-fc4-type>0x01 FCP FC-CT</active-fc4-type>
  <supported-speed>speed-8-gfc speed-16-gfc speed-32-gfc</supported-speed>
  <current-port-speed>speed-32-gfc</current-port-speed>
  <maximum-frame-size>2048</maximum-frame-size>
  <host-name>W16-68166</host-name>
  <node-name>20:00:00:10:1b:1f:1b:c1</node-name>
  <fabric-name>20:ad:c1:f1:1c:1a:ac:1c</fabric-name>
  <port-state>0x2</port-state>
  <number-of-discovered-ports>3</number-of-discovered-ports>
  <vsa-end-to-end-version>ELXE2EM:0001</vsa-end-to-end-version>
 </port>
```

```
</Response>
```

brocade-fibrechannel-configuration/chassis-config-settings

Description

Corresponds to the configurechassis command config parameters.

Parameters

Table 44: brocade-fibrechannel-configuration/chassis-config-settings

Name	Type	Description
firmware-synchronization-enabled	boolean	Enables the firmware from the active CP to automatically sync with the standby CP.
http-session-ttl	uint32	It denotes the timeout value for BNA, SANNav, REST, PyFOS. Units in seconds.
ezserver-enabled	boolean	To enable EZServer. True - enable False - disable.
cs-ctl-mode	enumeration	Describes about the CS_CTL mapping modes
sddq-limit	uint32	Specifies the number of user ports that can be quarantined in the chassis or unit. Valid values from 0 to 32. The default value is 10
vtap-qos-compatibility-mode	boolean	This variable enables or disables VTAP and QoS high priority zone compatibility mode. If VTAP and QoSH compatibility mode is enabled and affected VTAP features are also active on any of the logical switches within the chassis, then VTAP/QoS compatibility cannot be disabled. The VTAP features must be deactivated on all the logical switches in the chassis before disabling this setting. By default, this mode is disabled. true - ON, false - OFF.
secure-mode-enabled	boolean	Enables or Disables secure configuration download or upload.
file-suffix-enabled	boolean	If Enabled it appends a suffix to the uploaded configuration file.The suffix includes chassis name and timestamp in format yymmdd_hhmmss.
custom-index	uint32	OEM custom configuration. This attribute is for internal purpose only.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "chassis-config-settings": {
            "secure-mode-enabled": false,
            "file-suffix-enabled": false,
            "custom-index": 0,
            "firmware-synchronization-enabled": true,
            "http-session-ttl": 432000,
            "cs-ctl-mode": "default-mode",
            "sddq-limit": 10,
            "vtap-qos-compatibility-mode": false
        }
    }
}
```


PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fibrechannel-configuration/f-port-login-settings

Description

Provides parameters for F_Port login settings. Note that the switch must

be in a disabled state to configure F_Port login settings.

Parameters

Table 45: brocade-fibrechannel-configuration/f-port-login-settings

Name	Type	Description
max-logins	uint16	The maximum number of allowed logins for the switch. The maximum value is determined by multiplying 126 * number_of_ports_in_switch. The initial default value is 16 * number_of_ports_in_switch.
max-flogi-rate-per-switch	uint16	The maximum number of FLOGIs allowed on a switch.
stage-interval	uint16	The interval setting for the rate at which F_Ports are enabled.
free-fdisc	uint16	The number of logins allowed before staging. The values indicate: 0 : Disables the staging of FDISC logins. 1..255 : Enables staging and allows the specified number of logins. This parameter, if nonzero, enables staging of FDISC logins by rejecting the FDISC requests with logical busy, when the requests are more than the number of configured logins per second.

Name	Type	Description
enforce-login	uint16	The precedence for login when 2 devices with the same port WWN (PWWN) compete for login. All modes are for NPIV and non-NPIV F_Ports. The initial value is 0. The values indicate: 0 : The first FLOGI/FDISC login takes precedence, 1 : The second FLOGI/FDISC login takes precedence, 2 : The second FDISC login takes precedence on FDISC and on FLOGI login the first FLOGI takes precedence. This leaf is deprecated. Please use enforce-login-string.
enforce-login-string	enumeration	The precedence for login when 2 devices with the same port WWN (PWWN) compete for login. All modes are for NPIV and non-NPIV F_Ports. The initial value is first-flogi-or-first-fdisc-precedence.
max-flogi-rate-per-port	uint16	The maximum number of logins allowed, per second, on a given port. If the rate is exceeded, the port is fenced.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "f-port-login-settings": {
            "max-logins": 5120,
            "max-flogi-rate-per-switch": 0,
            "stage-interval": 0,
            "free-fdisc": 0,
            "enforce-login": 0,
            "enforce-login-string": "first-flogi-or-first-fdisc-precedence",
            "max-flogi-rate-per-port": 100
        }
    }
}
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fibrechannel-configuration/fabric

Description

Provides attributes applicable for a switch.

Parameters

Table 46: brocade-fibrechannel-configuration/fabric

Name	Type	Description
insistent-domain-id-enabled	boolean	Indicates whether Insistent Domain ID (IDID) is enabled. For a FICON switch, this leaf is read only and you cannot update the parameter. true : IDID is enabled. Switches that cannot be programmed with a unique domain ID will segment from the fabric. Check each switch that has IDID configured, and make sure that their domain IDs are unique within the configuration. false : IDID is disabled. The fabric automatically resolves domain ID conflicts during fabric merges or builds unless IDID is configured.
principal-selection-enabled	boolean	Enables or disables principal switch selection. true : Enables principal switch selection mode. false : Disables principal switch selection mode.
principal-priority	fibrechannel:brocade-hex-string-type	Sets the principal selection priority of the switch. Not all of these values can be assigned. This mechanism provide a preference for a switch requesting to be the principal switch in a fabric, but they do not provide an absolute guarantee that a switch requesting to be the principal switch is granted this status. 0x01 Highest and default priority (if priority nt given). 0x02 Should be avoided as this is internal priority given during fabric rebuild. 0x03- Priority value range. Higher numbers 0xFE mean lower priority. 0xFF Switch is not capable of acting as a principal switch.
preserved-domain-id-mode-enabled	boolean	Enables or disables Preserved Domain ID Allocation feature. true : Enables Preserved Domain ID Allocation feature. false : Disables Preserved Domain ID Allocation feature.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fibrechannel-configuration/port-configuration

Description

Port configuration parameters.

Parameters

Table 47: brocade-fibrechannel-configuration/port-configuration

Name	Type	Description
portname-mode	string	The available port name modes: off : No port name configuration. default : The port name corresponds to the port index. The following modes are not available in AG mode: fdmi : The port name corresponds to the attached FDMI host name. dynamic : The port name corresponds to the format defined by the dynamic-portname-format leaf.
dynamic-portname-format	string	The format of the dynamic port name is composed of fields that are mapped using the following characters: S : Switch name T : Port type I : Port index C : Slot number/port number (only applicable on the chassis) A : Alias name F : FDMI host name R : Remote switch name Multiple unique fields may be specified, separated by hyphens (-), periods (.), or underscores (_). Fields may not be repeated and field separators must be the same.
dynamic-d-port-enabled	boolean	The port has D_Port capability but it is not explicitly configured. Tests are always initiated by the remote side of the port. true : Dynamic D_Port configuration is enabled. false : Dynamic D_Port configuration is disabled.
on-demand-d-port-enabled	boolean	The port has D_Port capability, but it is not explicitly configured. The test is initiated by internal events of the switch, such as slotpoweroff/slotpoweron or persistentdisable/enable. The port remains a D_Port until the test is successful. true : On demand D_Port configuration is enabled. false : On demand D_Port configuration is disabled.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <port-configuration>
  <portname-mode>default</portname-mode>
  <dynamic-portname-format>S.T.I</dynamic-portname-format>
  <dynamic-d-port-enabled>true</dynamic-d-port-enabled>
  <on-demand-d-port-enabled>false</on-demand-d-port-enabled>
 </port-configuration>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fibrechannel-configuration/switch-configuration

Description

Provides switch configuration parameters.

Parameters

Table 48: brocade-fibrechannel-configuration/switch-configuration

Name	Type	Description
trunk-enabled	boolean	Enables or disables trunking on all ports in the logical switch. At the port level, you can disable trunking using the port-level REST method. Note that this operation is persistent and disruptive for all ports in the logical switch. true : Trunking is enabled on all ports in the logical switch. false : Trunking is disabled on all ports in the logical switch.
wwn-port-id-mode	boolean	Indicates whether WWN port ID mode is enabled on a switch. When enabled, WWN port ID mode supports both dynamic and static WWN-based PID assignment. Every time the device logs into the switch, the device is guaranteed the same PID.it is guaranteed to get the same PID. true : WWN port ID mode is enabled on the false : WWN port ID mode is disabled on the switch.
edge-hold-time	uint16	The maximum time that a frame can wait after it is received on the ingress port and before it is delivered to the egress port. Initial Value is Medium (220 ms). For a non-default logical switch, valid values include: 80 : Edge Hold Time, Low (80 ms), 220 : Edge Hold Time, Medium (220 ms), 500 : Edge Hold Time. High (500 ms). For a default switch the valid values can be: 80..500 : User-defined values are allowed only on default switch.
area-mode	uint8	The area mode for a default switch or non-default, non-FICON logical switch. Note that the switch must be in a disabled state to configure the area mode. Applicable for X6-4 and X6-8 chassis family. For a default switch, valid values include: 0 : 10-bit addressing mode (default) 1 : Zero-based area assignment (8-bit address mode) For a non-default and non-FICON logical switch, valid values include: 0 : 10-bit addressing mode (default) 1 : Zero-based area assignment 2 : Zero-based area assignment A non-default logical switch with FICON mode enabled is not supported. This leaf is deprecated. Please use area-mode-v2.

Name	Type	Description
area-mode-v2	enumeration	The area mode for a default switch or non-default, non-FICON logical switch. Note that the switch must be in a disabled state to configure the area mode. For a default switch, valid values include: 10-bit addressing mode (default) Zero-based area assignment (8-bit address mode) For a non-default and non-FICON logical switch, valid values include: 10-bit addressing mode (default) Zero-based area assignment Port-based area assignment A non-default logical switch with FICON mode enabled is not supported. This leaf is applicable for X6-4 and X6-8 chassis family.
xisl-enabled	boolean	Indicates whether XISL feature is enabled or disabled. This is mutually exclusive with the Application Header configuration. By default XISL feature is enabled on the switch Turning off XISL use requires confirmation because all LISLs will be removed upon execution. true : XISL use is enabled on the switch. false : XISL use is disabled on the switch.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "switch-configuration": {
            "area-mode": 0,
            "area-mode-v2": "10-bit-addressing-mode",
            "edge-hold-time": 220,
            "xisl-enabled": false
        }
    }
}
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fibrechannel-configuration/zone-configuration

Description

Zoning configuration parameters.

Parameters

Table 49: brocade-fibrechannel-configuration/zone-configuration

Name	Type	Description
node-name-zoning-enabled	boolean	Enables or disables node name checking for zoning. The switch must be disabled to change this value. true : Node name checking for zoning is enabled. false : Node name checking for zoning is disabled.
fabric-lock-timeout	uint16	Specifies the zone fabric lock timeout duration in minutes. The default value is 5 minutes.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "zone-configuration": {
            "node-name-zoning-enabled": true,
            "fabric-lock-timeout": 5
        }
    }
}
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fibrechannel-diagnostics/fibrechannel-diagnostics

Description

The list of ports on the device.

System-controlled interfaces created by the system are always present in this list, whether they are configured or not.

Parameters

Table 50: brocade-fibrechannel-diagnostics/fibrechannel-diagnostics

Name	Type	Description
name	fibrechannel:slot-port-name-type	The name of the interface. The slot and port number of the port (slot/port). Maximum of 12 slots and maximum of 1024 ports.
diagnostic-control	uint32	Manually terminate the test, re-initiate the test, or stop the test on a diagnostic port (D Port). 0 : Stop - terminates the diagnostics on the D Port 1 : Start - initiates the diagnostics on the D Port 2 : Restart - restarts the D Port diagnostic tests with the configured parameters This option is not supported for D Port HBA connections. NOTE: write-only leaf node and would not be present in GET request.
mode	enumeration	The testing mode. Indicates whether the test was started manually or automatically.
state	string	On the link initiator, this field displays the overall test status as PASSED, FAILED, FAILED (Protocol error), Unable to start/restart the test, SKIPPED, IN PROGRESS, NOT STARTED, or STOPPED. On the link responder, the status displays PASSED, FAILED, STOPPED, or RESPONDER. Please refer to the 'error-message' for the detailed error message.
error-message	string	If any, detailed error message for the overall test status is displayed.
distance	int32	The estimated physical distance of the inter-switch link in meters. The Link distance for SFPs or ICL ports which do not support E_WRAP and O_WRAP have a precision of +/- 50 meters for 8 Gbps LWL SFPs, 8 Gbps ELWL SFPs, 10 Gbps SFPs, ICL ports and QSFP ports, and +/- 5 meters for 16 Gbps, 32 Gbps and 64 Gbps SFPs. When FEC is active, precision is +/- 10 meters for all the 16 Gbps ports. -1 : indicates an unknown distance
remote-switch-wwn	fibrechannel:wwn-type	The WWN of the remote switch connected to this D-Port.
remote-port-index	fibrechannel:user-port-number-type	The user port index of the remote port connected to this D-Port.
start-time	string	Time diagnostics started.
frame-count	yang:counter32	Number of frames sent for link traffic test (default is 1 million frames). Note: This is mutually exclusive with the 'time' leaf.
frame-size	uint32	The test frame size in bytes (default is 1024 bytes). The size of the frames should be given in multiples of 4; otherwise the nearest higher multiple of 4 value will be taken as frame size.

Name	Type	Description
time	string	How long the test should run in the format hh:mm, where hh is hours and mm is minutes. Note: This is mutually exclusive with the 'frame-count' leaf.
rt-latency	uint32	The round trip link latency in nanoseconds or unknown when the test is in progress. Round trip link latency is the time required for the light to travel from a given port to the port at the remote side of the link and back again. Round trip latency for D_Port HBA connection is unknown when the HBA end does not have O_WRAP capability, in that case this leaf will return zero value.
buffers-required	string	The number of buffers required for the given distance and speed. Irrespective of test frame size, the number of buffers required is calculated for 2112-byte frames.
end-time	string	The date and time that the diagnostic test ended.
electrical-loopback-test	Container	An electrical loopback test; Supported only on 16 Gbps, 32 Gbps and 64 Gbps SFPs capable of electrical loopback.
- result	result-type	Result is the completion status of the test.
- comments	comments-type	Comments are additional textual information elaborating on the completion status when test is skipped or fails.
- start-time	fibrechannel:time-24hr-extended-type	Electrical loopback test start time.
- estimate-time	fibrechannel:time-24hr-extended-type	Estimated completion time of electrical loopback test.
optical-loopback-test	Container	An optical loopback test; Supported only on 16 Gbps, 32 Gbps and 64 Gbps SFPs capable of optical loopback.
- result	result-type	Result is the completion status of the test.
- comments	comments-type	Comments are additional textual information elaborating on the completion status when test is skipped or fails.
- start-time	fibrechannel:time-24hr-extended-type	Optical loopback test start time.
- estimate-time	fibrechannel:time-24hr-extended-type	Estimated completion time of optical loopback test.
link-traffic-test	Container	The link traffic test.
- result	result-type	Result is the completion status of the test.
- comments	comments-type	Comments are additional textual information elaborating on the completion status when test is skipped or fails.
- start-time	fibrechannel:time-24hr-extended-type	Link traffic test start time.
- estimate-time	fibrechannel:time-24hr-extended-type	Estimated completion time of link traffic test.
payload-pattern	Container	The user-defined payload pattern.

Name	Type	Description
- pattern	string	The name of the predefined industry pattern names used for frame traffic test. The valid patterns are: BYTE_NOT QUAD_NOT WORD_NOT BYTE_RAMP QUAD_RAMP WORD_RAMP BYTE_LFSR RANDOM CRPAT CSPAT CHALF_SQ CQTR_SQ RDRAM_PAT jCRPAT jCJTPAT jCSPAT PRED_RAND SMI_TEST CJPAT QUAD_NOTP JSPAT JTSPAT
- payload	string	Specifies the user defined payload . Payload should be in the range of 0x1-0xffffffff.
fec	Container	Enables Forward Error Correction (FEC) for diagnostic testing. This option is not supported on ports configured with Dense Wavelength Division Multiplexing (DWDM)
- enable	enumeration	When FEC is enabled on a port, the sender adds systematically generated error-correcting code (ECC) to its data transmission. This mechanism allows the receiver to detect and correct errors without needing to get additional information from the sender. This leaf only supports enabling FEC.
- active	enumeration	The FEC state can be active or inactive. An active FEC state indicates that FEC is enabled and actually running. An inactive state can indicate two conditions: FEC is enabled, but not running due to some error condition (for example, FEC may not be enabled on both ends of the link). Or FEC is disabled and therefore inactive.
- option	enumeration	The option indicates that D-Port diagnostic test is using FEC feature or not.
cr	Container	The Credit Recovery (CR) status.
- enable	enumeration	The Credit Recovery enables credits or frames to be recovered. This leaf only supports enabling Credit Recovery.
- active	enumeration	Credit Recovery is active.
- option	enumeration	The leaf option indicates whether the D-Port diagnostic test is using Credit Recovery (CR).
failure-report	Container	Displays details on the local port and remote port errors, if any,

Name	Type	Description
- errors-detected-local	enumeration	Displays details on the local port, if any. Deprecated: Please use 'failure-report-local-details' leaf-list instead.
- errors-detected-remote	enumeration	Displays details on the remote port, if any. Deprecated: Please use 'failure-report-remote-details' leaf-list instead.
failure-report-local-errors	Container	Displays details on the local port errors.
- error	error-stats-type	Displays error stats name on the local port, if any.
failure-report-remote-errors	Container	Displays details on the remote port errors.
- error	error-stats-type	Displays error stats name on the remote port, if any.
egress-power-loss	Container	The power values and power loss in the transmit path. The following values are displayed: power transmitted from the local side of the link (Tx) power received on the remote side of the link (Rx), the power loss, that is, the relative difference in power (Diff), and the status of the power loss (No Loss, within tolerable limits, and outside tolerable limits).
- tx	decimal64	Power transmitted from the local side of the link (Tx). In this case, if D-port test is failing or not yet started Tx value might not be available. In such case this leaf is not displayed.
- rx	decimal64	Power received on the remote side of the link (Rx). In this case, if D-port test is failing or not yet started Rx value might not be available. In such case this leaf is not displayed.
- loss	decimal64	The power loss, that is, the relative difference in power (Diff). This leaf is not displayed if the value is not available.
- comments	string	Status of the power loss (No Loss, within tolerable limits, and outside tolerable limits). This leaf returns an empty string if the value is not available.
ingress-power-loss	Container	The power values and power loss in the receive path. The following values are displayed: power received on the local side of the link (Rx), power transmitted from the remote side of the link (Tx), the power loss, that is, the relative difference in power (Diff), and the status of the power loss (No Loss, within tolerable limits, and outside tolerable limits).
- tx	decimal64	Power transmitted from the remote side of the link (Tx). In this case, if D-port test is failing or not yet started Tx value might not be available. In such case this leaf is not displayed.
- rx	decimal64	Power received on the local side of the link (Rx). In this case, if D-port test is failing or not yet started Rx value might not be available. In such case this leaf is not displayed.
- loss	decimal64	The power loss, that is, the relative difference in power (Diff). This leaf is not displayed if the value is not available.

Name	Type	Description
- comments	string	Status of the power loss (No Loss, within tolerable limits, and outside tolerable limits). This leaf returns an empty string if the value is not available.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "fibrechannel-diagnostics": [
            {
                "name": "7/10",
                "diagnostic-control": 0,
                "mode": "automatic",
                "state": "NOT STARTED",
                "error-message": "",
                "remote-switch-wwn": "10:00:c4:f5:7c:d3:be:ae",
                "remote-port-index": 3,
                "link-traffic-test": {
                    "result": "NOT STARTED",
                    "comments": "",
                    "start-time": "07:33:25",
                    "estimate-time": ""
                },
                "start-time": "Wed Mar  1 07:33:22 2023",
                "frame-count": 1,
                "frame-size": 1024,
                "fec": {
                    "enable": "yes",
                    "active": "no",
                    "option": "no"
                },
                "cr": {
                    "enable": "yes",
                    "active": "no",
                    "option": "no"
                },
                "rt-latency": 0,
                "buffers-required": "",
```

```
                "failure-report": {
                    "errors-detected-local": "no",
                    "errors-detected-remote": "no"
                },
                "failure-report-local-errors": {
                    "error": [
                        "none"
                    ]
                },
                "failure-report-remote-errors": {
                    "error": [
                        "none"
                    ]
                },
                "end-time": "Wed Mar  1 07:33:25 2023",
                "egress-power-loss": {
                    "tx": -4.00,
                    "rx": -6.10,
                    "loss": 2.10,
                    "comments": "Cable power loss is within tolerable limit"
                },
                "ingress-power-loss": {
                    "tx": -2.80,
                    "rx": -3.20,
                    "loss": 0.40,
                    "comments": "Cable power loss is within tolerable limit"
                },
                "payload-pattern": {
                    "pattern": "jCRPAT"
                }
            }
        ]
    }
}
```


PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fibrechannel-logical-switch/fibrechannel-logical-switch

Description

The logical switch configuration.

Parameters

Table 51: brocade-fibrechannel-logical-switch/fibrechannel-logical-switch

Name	Type	Description
fabric-id	uint32	The virtual fabric identification (VFID) of the logical switch.
switch-wwn	fibrechannel:wwn-type	The switch WWN, which is a unique numeric identifier for the switch assigned by the chassis.
base-switch-enabled	uint8	The logical switch is the base switch in the chassis. This can be specified during creation of logical switch. This cannot be specified during deletion of logical switch. 0 : disabled 1 : enabled Deprecated: Please use 'base-switch-enabled-v2' instead.
base-switch-enabled-v2	boolean	The logical switch is the base switch in the chassis. This can be specified during creation of logical switch. This cannot be specified during deletion of logical switch. false : disabled true : enabled
default-switch-status	uint8	The logical switch is the default switch in the chassis. 0 : disabled 1 : enabled Deprecated: Please use 'default-switch' instead.
default-switch	boolean	Indicates whether the logical switch is the default switch in the chassis. true : The logical switch is the default switch in the chassis. false : The logical switch is not the default switch in the chassis.
logical-isl-enabled	uint8	Enables logical ISLs (LISLs). Disabling LISLs is allowed only during logical switch creation. Enabling LISLs is not allowed on a base switch. 0 : disabled 1 : enabled Deprecated: Please use 'logical-isl-enabled-v2' instead.
logical-isl-enabled-v2	boolean	Enables logical ISLs (LISLs). Disabling LISLs is allowed only during logical switch creation. Enabling LISLs is not allowed on a base switch. false : disabled true : enabled
ficon-mode-enabled	uint8	Indicates the FICON mode. The values are: 0 : disabled 1 : enabled Disabling FICON mode in a logical switch is not allowed. Deprecated: Please use 'ficon-mode-enabled-v2' instead.
ficon-mode-enabled-v2	boolean	Indicates the FICON mode. The values are: false : disabled true : enabled Disabling FICON mode in a logical switch is not allowed.
port-member-list	Container	Contains list of Logical Switch port interface names.

Name	Type	Description
- port-member	fibrechannel:slot-port-name-type	The name of the fibre channel interface. The slot and port number of the port (slot/port). This list only contains fibre channel ports and flex ports.
ge-port-member-list	Container	Contains list of Logical Switch port interface names.
- port-member	fibrechannel:slot-port-name-type	The name of the Gigabit Ethernet (GE) interface. The slot and port number of the port (slot/port). This list only contains Gigabit Ethernet ports.
port-index-members	Container	Contains list of Logical Switch port index numbers.
- port-index	fibrechannel:user-port-number-type	The fibre channel port index number. This list only contains fibre channel port and flex port indexes.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-fibrechannel-routing/edge-fabric-alias

Description

Displays edge fabric id's corresponding alias name

Parameters

Table 52: brocade-fibrechannel-routing/edge-fabric-alias

Name	Type	Description
edge-fabric-id	fibrechannel:fabric-id-type	The ID of the fabric in which the LSAN zone was created
alias-name	string	Alias name of the specific edge fabric id.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-fibrechannel-routing/fibrechannel-router

Description

Displays the list of FC routers and its information that exists in the

FC Router backbone fabric.

This list supports the following query parameter combination:

domain-id

Parameters

Table 53: brocade-fibrechannel-routing/fibrechannel-router

Name	Type	Description
domain-id	fibrechannel:domain-id-type	The Domain ID of the FC Router. This leaf can be queried.
router-wwn	fibrechannel:wwn-type	The World Wide Name of the FC Router.
ipv4-address	union {inet:ipv4-address;fibrechannel:brocade-empty-string-type;}	The Ethernet IPv4 address of the FC Router.
ipv6-address	union {inet:ipv6-address;fibrechannel:brocade-empty-string-type;}	The Ethernet IPv6 address of the FC Router.
switch-user-friendly-name	string	Switch name of the FC Router.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-router>
        <domain-id>43</domain-id>
        <router-wwn>10:00:c1:f1:1c:1a:ac:1c</router-wwn>
        <ipv4-address>10.10.10.103</ipv4-address>
        <ipv6-address>2120:100:1:fa01::fb10</ipv6-address>
        <switch-user-friendly-name>X6-4</switch-user-friendly-name>
    </fibrechannel-router>
</Response>
```

brocade-fibrechannel-routing/lsan-device

Description

Displays the information of each device.

Parameters

Table 54: brocade-fibrechannel-routing/lsan-device

Name	Type	Description
source-fabric-id	fibrechannel:fabric-id-type	The fabric in which the device are physically present.
device-wwn	fibrechannel:wwn-type	The WWN of the device.
proxy-pid	fibrechannel:fcid-hex-string-type	The proxy pid of the device.
imported-fabric-id	fibrechannel:fabric-id-type	The fabric in which the devices are imported.
physical-pid	fibrechannel:fcid-hex-string-type	The port ID of the device in source fabric.
device-status	fcr-device-state	States include: configured - Device is configured to be in an LSAN, but the device is neither imported nor exists in this fabric initializing - Device is in an intermediate state. It is not yet imported into the fabric. exist - Device exists in this fabric (the fabric of the zone entry). imported - Device has been imported (proxy created) into this fabric.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <lsan-device>
  <source-fabric-id>100</source-fabric-id>
  <device-wwn>10:00:00:0e:1e:11:01:01</device-wwn>
  <physical-pid>0xec0500</physical-pid>
  <device-status>exist</device-status>
 </lsan-device>
 <lsan-device>
  <source-fabric-id>100</source-fabric-id>
  <device-wwn>21:00:00:0e:1e:12:02:02</device-wwn>
  <physical-pid>0x2baa00</physical-pid>
  <device-status>exist</device-status>
```

```
  </lsan-device>
 ...

  <lsan-device>
   <source-fabric-id>100</source-fabric-id>
   <device-wwn>23:00:00:3b:30:13:d3:03</device-wwn>
   <physical-pid>0xeb0001</physical-pid>
   <proxy-pid>0xcbf401</proxy-pid>
   <imported-fabric-id>128</imported-fabric-id>
   <device-status>imported</device-status>
  </lsan-device>
 </Response>
```

brocade-fibrechannel-routing/lsan-zone

Description

Displays the list of LSAN zone members in a particular zone.

Parameters

Table 55: brocade-fibrechannel-routing/lsan-zone

Name	Type	Description
edge-fabric-id	fibrechannel:fabric-id-type	The ID of the fabric in which the LSAN zone was created
zone-name	zone:zoning-name-type	The name of the zone
members	Container	Contains the list of LSAN zone members of a particular zone
- member	fibrechannel:wwn-type	The member of the zone

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <lsan-zone>
  <edge-fabric-id>100</edge-fabric-id>
  <zone-name>lsan_68162_Violin</zone-name>
```

```
   <members>
    <member>10:00:1c:2c:ff:14:ec:00</member>
    <member>10:00:1c:2c:ff:14:ec:01</member>
    <member>10:00:1c:2c:ff:14:d8:03</member>
    <member>21:00:00:1b:17:18:da:f4</member>
   </members>
  </lsan-zone>
 ...
  <lsan-zone>
   <edge-fabric-id>128</edge-fabric-id>
   <zone-name>lsan_68163_Violin</zone-name>
   <members>
    <member>10:00:00:00:c9:f8:04:05</member>
    <member>10:00:00:00:c9:f8:04:06</member>
    <member>21:00:00:1b:10:11:d8:07</member>
    <member>21:00:00:1b:10:11:da:f8</member>
   </members>
  </lsan-zone>
  <lsan-zone>
   <edge-fabric-id>128</edge-fabric-id>
   <zone-name>lsan_68164_Violin</zone-name>
   <members>
    <member>21:00:00:0e:1e:07:03:09</member>
    <member>21:00:00:0e:1e:07:04:10</member>
    <member>21:00:00:1b:10:11:d8:11</member>
    <member>21:00:00:1b:10:11:da:f2</member>
   </members>
  </lsan-zone>
 </Response>
```

brocade-fibrechannel-routing/proxy-config

Description

Displays or configures proxy devices presented by an FC Router.

Parameters

Table 56: brocade-fibrechannel-routing/proxy-config

Name	Type	Description
imported-fabric-id	fibrechannel:fabric-id-type	The fabric in which the devices are imported.
device-wwn	fibrechannel:wwn-type	The WWN of the device.
proxy-device-slot	fibrechannel:area-address-type	The slot used for the device WWN. The device WWN-to-slot association can be configured persistently. The slot format is 0xAAPP, where AA specifies the translate domain area_ID (0xF0..0xFF) and 0xPP specifies the Port ID value or the low 8-bits of the proxy device address (0x01..0xFE).

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-fibrechannel-routing/router-statistics

Description

Information about the resources that is available in the fibrechannel

router.

Parameters

Table 57: brocade-fibrechannel-routing/router-statistics

Name	Type	Description
lsan-zones-in-use	uint32	Displays the count of currently used LSAN zones.
maximum-lsan-devices	uint32	Maximum available LSAN device database entries.
lsan-devices-in-use	uint32	Currently used LSAN device database entries. Each proxy or physical device constitutes an entry.
maximum-proxy-device-slots	uint32	Maximum available proxy device slots. A proxy device is presented to an edge fabric as being connected to a translate domain slot. A slot is the port number and AL_PA combination. The slot-to-device WWN association is persistently stored.

Name	Type	Description
proxy-device-slots-in-use	uint32	Currently used count of proxy device slots.
maximum-proxy-devices	uint32	Maximum count of proxy device for the physical port(EX_Port).
maximum-nr-ports	uint32	Displays maximum available NR_Port entries. Destination NR_Port entries are stored at every physical port for routing decision purposes.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <router-statistics>
        <lsan-zones-in-use>2</lsan-zones-in-use>
        <maximum-lsan-devices>15000</maximum-lsan-devices>
        <lsan-devices-in-use>12</lsan-devices-in-use>
        <maximum-proxy-device-slots>15000</maximum-proxy-device-slots>
        <proxy-device-slots-in-use>6</proxy-device-slots-in-use>
        <maximum-proxy-devices>4000</maximum-proxy-devices>
        <maximum-nr-ports>1000</maximum-nr-ports>
    </router-statistics>
</Response>
```

brocade-fibrechannel-routing/routing-configuration

Description

Generic FCR configuration parameters.

Parameters

Table 58: brocade-fibrechannel-routing/routing-configuration

Name	Type	Description
maximum-lsan-count	uint32	The default maximum LSAN count is set to 3000. User can modify this LSAN count value. The valid values are 3000, 5000 and 7500. For Brocade 7810, the maximum LSAN zone count is 3000.
backbone-fabric-id	fibrechannel:fabric-id-type	Specifies the Backbone Fabric ID. It uniquely identifies a fabric in FC Router configurations. The backbone fabric is the fabric attached to the U_Ports of the switch, for example, E_Ports or F_Ports. The backbone fabric ID must be unique across all fabrics connected to the FC Router. Backbone Fabric ID cannot be modified when virtual fabric mode is ON.
shortest-ifl	boolean	Enables or disables the shortest IFL mode in FC Router. When the shortest IFL mode is enabled, FC Router can choose a lowest-cost IFL path in the backbone fabric. TRUE : Enables the shortest IFL mode FALSE : Disables the shortest IFL mode
lsan-speed-tag	string	Allows the FCR to always import these target devices to the hosts specified in the LSANs that match the speed tag. Only one speed tag is allowed per FC router. Empty tags are not allowed. Not allowed if base-switch-enabled state false when vf-enabled state is true.
migration-mode	boolean	Enables or disables the migration mode on an FCR switch. Enabling the migration mode on an FCR facilitates the user to proceed with decommissioning actions leading to removal of the FCR switch from the backbone fabric. The decommission actions should result in disabling all the EX_Port(s) connected to the edge fabric from which traffic has been migrated (to other FCRs). Later, the user could disable the migration mode and/or physically remove the switch from the topology. Migration of FCR B2E traffic is not supported. true : Migration mode is enabled false : Migration mode is disabled
persistent-translate-domain-enabled	boolean	Enables or disables the state of persistence for translate domain ID configuration. FCR creates the translate domain for a remote fabric if a valid persistent translate domain ID is configured in the local fabric, even if no devices are imported or exported across the edge fabrics. Disabling the persistent translate domain parameter prevents the translate domain from being created. true : Persistent translate domain is enabled. false : Persistent translate domain is disabled. By default, persistent translate domain is enabled.
backbone-fabric-alias	string	Backbone Fabric's alias name.
lsan-enforce-tags	Container	Contains list of lsan enforce tags.

Name	Type	Description
- tag	string	Accepts only the LSANs from the edge fabric that matches the specified tag string into the local FCR database. A valid tag is a string of a maximum of eight characters. The maximum configurable enforce tags is eight. In case speed tag is configured, only seven enforce tags can be configured. Empty tags are not allowed. Configuration is not allowed if base-switch-enabled state false when vf-enabled state is true.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "routing-configuration": {
            "maximum-lsan-count": 3000,
            "backbone-fabric-id": 128,
            "shortest-ifl": false,
            "lsan-enforce-tags": {},
            "lsan-speed-tag": "",
            "migration-mode": false,
            "persistent-translate-domain-enabled": true,
            "backbone-fabric-alias": ""
        }
    }
}
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-fibrechannel-routing/stale-translate-domain

Description

Displays stale translate domains associated with the specific FID.

Only supports delete method.

Parameters

Table 59: brocade-fibrechannel-routing/stale-translate-domain

Name	Type	Description
imported-fabric-id	fibrechannel:fabric-id-type	The fabric in which the devices are imported.
stale-translate-domain-id	fibrechannel:domain-id-type	A translate domain becomes stale when the remote edge fabric for which this translate domain was created in the specified edge fabric becomes unreachable.
owner-front-domain-id	fibrechannel:domain-id-type	The owner front domain ID of the stale translate domain.
front-domain-owner	fibre-channel-router-owner	Whether the 'owner-front-domain-id' belongs to this FCR or the other remote FCR.
dual-backbone	boolean	Whether the topology is dual Backbone. true : Is a dual Backbone fabric. false : Not a dual Backbone fabric. NOTE: Translate domain can be owned by front domain of other Backbone. So, the 'stale-translate-domain-id' displayed may or may not be a Stale translate domain. To confirm, the stale entries should be verified in the other Backbone also.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fibrechannel-routing/translate-domain-config

Description

Displays the translate domain configuration.

Parameters

Table 60: brocade-fibrechannel-routing/translate-domain-config

Name	Type	Description
imported-fabric-id	fibrechannel:fabric-id-type	The fabric ID of the fabric that contains the translate domain.
exported-fabric-id	fibrechannel:fabric-id-type	The fabric ID of the remote fabric represented by this translate domain.
preferred-translate-domain-id	fibrechannel:domain-id-type	The preferred domain ID of the translate phantom domain. During a fabric build, he principal switch attempts to provide the translate domain with the requested domain ID, but it may not provide it if there are domain ID conflicts with other domains in the fabric. If the requested domain ID (such as the preferred domain ID) is unavailable, the domain ID assignment is completely at the discretion of the principal switch. The assignment domain ID is persistently stored and is used as the preferred domain ID in the future.
owner-front-domain-id	fibrechannel:domain-id-type	The owner front domain ID for this translate domain. This data is available only if the owner FC Router was nominated for the edge fabrics represented by 'imported-fabric-id' and 'exported-fabric-id.'
translate-wwn	fibrechannel:wwn-type	The translate WWN of the device. This data is available only if the translate domain is created by the FC Router, for the edge fabrics represented by 'imported-fabric-id' and 'exported-fabric-id.'
front-domain-owner	fibre-channel-router-owner	Whether the 'owner-front-domain-id' belongs to this FCR, the other FCR in the same Backbone fabric or the FCR in another Backbone fabric.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
```

```
<Response>
    <translate-domain-config>
        <imported-fabric-id>77</imported-fabric-id>
        <exported-fabric-id>128</exported-fabric-id>
        <preferred-translate-domain-id>200</preferred-translate-domain-id>
        <owner-front-domain-id>160</owner-front-domain-id>
        <translate-wwn>5c:4f:32:c1:aa:d1:ef:c2</translate-wwn>
        <front-domain-owner>local</front-domain-owner>
    </translate-domain-config>
    <translate-domain-config>
        <imported-fabric-id>128</imported-fabric-id>
        <exported-fabric-id>77</exported-fabric-id>
        <preferred-translate-domain-id>200</preferred-translate-domain-id>
        <owner-front-domain-id>160</owner-front-domain-id>
        <translate-wwn>5c:4f:32:c1:aa:d1:ef:c2</translate-wwn>
        <front-domain-owner>local</front-domain-owner>
    </translate-domain-config>
</Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-fibrechannel-switch/fibrechannel-switch

Description

The switch state parameters. Requests can be made using a query that

specifies the vf-id of the fabric. This request also provides the

switch state parameters.

Parameters

Table 61: brocade-fibrechannel-switch/fibrechannel-switch

Name	Type	Description
name	fibrechannel:wwn-type	The switch world wide name (WWN).
domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
user-friendly-name	string	The ASCII name assigned to the switch by the administrator.

Name	Type	Description
fcid	fibrechannel:fcid-type	The destination ID (D_ID) of the switch (decimal format). Deprecated: Use the 'fcid-hex' leaf instead.
fcid-hex	fibrechannel:fcid-hex-string-type	The destination id (D_ID) of the switch (hexadecimal format).
vf-id	int16	This is the virtual fabric identifier (VFID) of the logical switch. -1 : Indicates that the Virtual Fabric feature is unsupported or disabled.
principal	uint8	The principal switch in the fabric. The valid values are: 0 : This switch is not the principal switch in the fabric. 1 : This switch is the principal switch in the fabric.
is-principal	boolean	The principal switch in the fabric. true : This switch is the principal switch in the fabric. false : This switch is not the principal switch in the fabric.
enabled-state	uint32	The current state of the switch. The valid values are: 0 : Undefined. 2 : Switch is enabled. 3 : Switch is disabled. 7 : Switch is being tested. Note that the enabled-state can be set to 2 (enabled) or 3 (disabled) only; it cannot be set to 0 (undefined) or 7 (testing). Deprecated: Use the is-enabled-state leaf to configure the switch state and operational-status leaf to get the switch status.
is-enabled-state	boolean	The current configuration of the switch. true : The switch is enabled. false : The switch is disabled.
operational-status	uint32	The current state of the switch. The valid values are: 0 : Undefined. 2 : The switch is enabled. 3 : The switch is disabled. 7 : The switch is being tested.
operational-status-string	operational-status-type	The current state of the switch.
up-time	yang:timeticks	The period of time elapsed since last reboot of the specified switch. This value represents an absolute time interval for which the switch has been up, not the time measured between the last reboot and midnight, January 1, 1970 UTC, and the most recent reboot of the switch. This leaf is depricated, please use brocade-chassis/chassis-uptime.
model	string	The model name of the switch.
firmware-version	string	A human readable string identifying the firmware version running on the switch.

Name	Type	Description
subnet-mask	inet:ipv4-address	The IPv4 subnet mask of the switch IP network. In PATCH operation, manually configured or static - IPv4 subnet mask is written when DHCP is disabled. - IPv4 subnet mask is ignored when DHCP is enabled. Deprecated: Use the 'static-subnet-mask' and 'dhcp-subnet-mask' leafs in brocade-management-ip-interface instead.
ipfc-address	union {inet:ipv4-address;fibrechannel:brocade-empty-string-type;}	The IPv4 Fibre Channel address for the logical fabric IPFC network interface when AG mode is not enabled.
ipfc-subnet-mask	union {inet:ipv4-address;fibrechannel:brocade-empty-string-type;}	The subnet mask of the logical fabric IPFC network interface when AG mode is not enabled.
domain-name	string	The DNS domain name of the switch. The DNS domain name can be 4 to 63 alphanumeric characters. This DNS domain name must start with an alphabetic character, and cannot contain blank spaces and control characters. Passing a null string clears the domain name.
fabric-user-friendly-name	string	An ASCII string which displays the configured fabric name that applies to all the switches within the same fabric. The fabric name must be unique for each logical switch within a chassis.
ag-mode	uint8	The Access Gateway (AG) mode capability and state. A switch is capable of AG mode support when this value is non-zero. 0 : AG mode is not supported by this device. 1 : AG mode is supported and is currently disabled. 3 : AG mode is supported and is currently enabled. Deprecated: For write operations, use the 'ag-mode-string' leaf instead. For read operations, this leaf will be returned in responses with valid data comparable to the 'ag-mode-string' leaf. In the event of a write operation containing both the deprecated 'ag-mode' leaf as well as the 'ag-mode-string' leaf, both leafs must convey the same type. If they do not convey the same type, then the write operation will result in an error.
ag-mode-string	fibrechannel:ag-mode-type	The Access Gateway (AG) mode capability and state. A switch is capable of AG mode support when this value is not not-supported.
banner	string	A text string of alphanumeric characters. The banner displays during the log on process. The banner text can be 0 to 1022 characters. If the banner text exceeds the maximum limit, the software truncates the input.
in-order-delivery-enabled	boolean	Whether the In Order Delivery (IOD) feature is enabled on the switch. true: IOD is enabled. false: IOD is disabled.

Name	Type	Description
dynamic-load-sharing	fibrechannel:dynamic-load-sharing-level-type	The Dynamic Load Sharing (DLS) mode in use by the switch. capability is in use by the switch. The valid values are: disabled : Disables DLS. dls : Enables the DLS capablility only. lossless-dls : Enables DLS with Lossless capablity. two-hop-lossless-dls : Enables DLS with both variants of Lossless. Note: When the Advanced Performance Tuning policy for the switch is Device-Based or Exchange-Based, then the DLS setting cannot be disabled.
advanced-performance-tuning-policy	fibrechannel:advanced-performance-tuning-policy-type	The current Advanced Performance Tuning (APT) policy for the local switch. The APT policy controls how the switch creates the routes that are used by traffic flowing through the switch. The valid values are: port-based : Port-Based Routing mode device-based : Device-Based Routing mode exchange-based : Exchange-Based Routing mode
lacp-system-priority	uint32	The system priority of this switch. This priority is used for determining the system that is responsible for resolving conflicts in the choice of aggregation groups. A lower numerical value has a higher priority.
lacp-system-mac-address	ietf:mac-address	The LACP system MAC address of the switch.
switch-persistent-enabled	boolean	Whether the switch is in a persistently enabled state. true : The switch is persistently enabled. false : The switch is persistently disabled.
switch-addressing-mode	switch-addressing-mode-type	The fabric address mode value.
switch-beacon-enabled	boolean	The switch beaconing state. true : Switch beaconing is enabled. false : Switch beaconing is disabled.
high-integrity-fabric-mode-enabled	boolean	The high intensity fabric mode of the switch. true : The switch has high intensity fabric mode enabled. false : The switch has high intensity fabric mode disabled.
ip-address	Container	A list of out-of-band IP addresses assigned to the Ethernet port on the switch. This container is needed by underlying infrastructure. In PATCH operation, manually configured or static - IPv4 address is written when DHCP is disabled. - IPv4 address is ignored when DHCP is enabled. - IPv6 address is written if not already existing as a DHCPv6 or stateless address. - IPv6 address causes duplicate address error if already existing as a DHCPv6 or stateless address. Deprecated: Use the static-ip-addresses and stateful-ip-addresses leafs in brocade-management-ip-interface instead.
- ip-address	union {inet:ip-address;inet:ip-prefix;}	A list of out-of-band IP addresses assigned to the Ethernet port of the switch.
ip-static-gateway-list	Container	A list of IPv4 and IPv6 static gateway addresses for the switch IP address.

Name	Type	Description
- ip-static-gateway	fru:ip-gateway-type	A list of static gateway IP addresses of an IP router that can route packets to the destination IP address. The gateway address must be on the same IP subnet as one of the port IP addresses. Only one IPv4 static gateway and one IPv6 static gateway are allowed. In a PATCH operation, manually configured or static - IPv4 gateway is written when DHCP is disabled. - IPv4 gateway is ignored when DHCP is enabled. - IPv6 gateway is written if not already existing as a stateless gateway. - IPv6 gateway causes duplicate gateway error if already existing as a stateless gateway. Deprecated: Use the 'static-ip-gateways', 'dhcp-gateway' and 'stateless-ip-gateways' leafs in brocade-management-ip-interface instead.
dns-servers	Container	A list of DNS server addresses.
- dns-server	union {inet:ip-address;inet:ip-prefix;}	A list of DNS server addresses which can handle the mapping of the domain names to IP address of an internet resources which is needed by various networking protocols. Brocade FC switches support maximum of two DNS servers.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <fibrechannel-switch>
  <name>10:00:c4:f5:7c:4a:ac:6c</name>
  <domain-id>43</domain-id>
  <fcid>16776235</fcid>
  <fcid-hex>0xfffc2b</fcid-hex>
  <user-friendly-name>66-243_X6-4_Venator</user-friendly-name>
  <enabled-state>2</enabled-state>
  <is-enabled-state>true</is-enabled-state>
  <operational-status>2</operational-status>
  <banner/>
  <up-time>12766</up-time>
  <domain-name>brm.bsnlab.broadcom.net</domain-name>
  <dns-servers>
```

```
    <dns-server>192.19.189.10</dns-server>
    <dns-server>192.19.189.20</dns-server>
   </dns-servers>
   <principal>0</principal>
   <ip-address>
    <ip-address>10.38.66.243</ip-address>
   </ip-address>
   <subnet-mask>255.255.240.0</subnet-mask>
   <model>165.0</model>
   <firmware-version>v9.0.0_bld85</firmware-version>
   <ip-static-gateway-list>
    <ip-static-gateway>10.38.64.1</ip-static-gateway>
   </ip-static-gateway-list>
   <vf-id>-1</vf-id>
   <fabric-user-friendly-name>East</fabric-user-friendly-name>
   <ag-mode>0</ag-mode>
   <dynamic-load-sharing>lossless-dls</dynamic-load-sharing>
   <in-order-delivery-enabled>false</in-order-delivery-enabled>
   <advanced-performance-tuning-policy>exchange-based</advanced-performance-tuning-policy>
   <lacp-system-mac-address>c4:f5:7c:59:ac:6e</lacp-system-mac-address>
   <lacp-system-priority>32768</lacp-system-priority>
  </fibrechannel-switch>
 </Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fibrechannel-switch/switch-fabric-statistics

Description

The statistics data for the fabric incidents.

Parameters

Table 62: brocade-fibrechannel-switch/switch-fabric-statistics

Name	Type	Description
fabric-incident	fabric-incident-type	The incidents that occurred in a fabric such as domain ID forcibly changed, E_Port offline transitions, fabric reconfigurations, and fabric segmentations.
segmentation-reason	segmentation-reason-type	The reason for the fabric segmentation.
last-event	boolean	Whether the fabric incident is the last event that occurred.
incident-count	yang:zero-based-counter32	The total number of times the specific event occurred on various ports on the switch.
latest-port-index	fibrechannel:port-number-type	The port index where the latest incident occurred.
time-stamp	yang:date-and-time	The date and time when the latest incident occurred.

Produces

- application/xml
- application/json

brocade-fibrechannel-switch/topology-domain

Description

The topology data for one of the fabric domains, including the Shortest Path First (SPF) path cost (metric), the total number of potential paths that qualify based on the SPF calculations, and a list of the egress (out-ports) that are in use for routing traffic to the domain.

Parameters

Table 63: brocade-fibrechannel-switch/topology-domain

Name	Type	Description
domain-id	fibrechannel:domain-id-type	The fabric domain for which this topology data applies.
path-count	yang:gauge32	The number of in-use paths, that have the same cost (metric), for routing traffic that is destined for the domain.
metric	uint32	The path cost for routing traffic to the domain.
is-local-translate-domain	boolean	Whether the domain is a local translate domain, meaning the domain is a translate domain, and it is directly connected to this FC Router switch. true : The domain is a local translate domain. false : The domain is not a local translate domain.
out-ports	Container	A list of egress ports that are programmed for use when routing traffic to use the domain.
- port	fibrechannel:user-port-number-type	The out-port to reach destination domain.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <topology-domain>
        <domain-id>92</domain-id>
        <path-count>4</path-count>
        <metric>500</metric>
        <is-local-translate-domain>false</is-local-translate-domain>
        <out-ports>
            <port>128</port>
            <port>129</port>
            <port>159</port>
            <port>160</port>
        </out-ports>
    </topology-domain>
...
    <topology-domain>
        <domain-id>239</domain-id>
        <path-count>2</path-count>
        <metric>1000</metric>
        <is-local-translate-domain>false</is-local-translate-domain>
        <out-ports>
            <port>132</port>
            <port>167</port>
        </out-ports>
    </topology-domain>
</Response>
```

brocade-fibrechannel-switch/topology-error

Description

A list of any topology errors as detected by FSPF.

If the fabric is building or is unstable, this list may be empty.

Parameters

Table 64: brocade-fibrechannel-switch/topology-error

Name	Type	Description
domain-id	fibrechannel:domain-id-type	The domain ID for the error entry.
error-type	fibrechannel:topology-report-error-type	The type of topology error for the entry.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-fibrechannel-switch/topology-route

Description

A route currently programmed by the local switch. Includes
one of the egress (out-ports), any ingress (in-ports), and
the number of hops to reach the domain, by the switch.

Parameters

Table 65: brocade-fibrechannel-switch/topology-route

Name	Type	Description
domain-id	fibrechannel:domain-id-type	The domain ID of the switch to which this route is destined.
hop-count	uint32	The greatest number of hops (across all capable paths) to reach the domain as of the latest Shortest Path First calculations. Note: The hop count is per destination domain, as opposed to per path. So all out-ports report the same number of hops.
out-port	fibrechannel:user-port-number-type	The egress (out-port) to reach the domain
in-ports	Container	A list of the user-ports that are ingress (in-ports) using this out-port for any traffic destined to the identified domain ID.
- port	fibrechannel:user-port-number-type	A user-port routed to the identified out-port for any frames destined for the identified domain.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <topology-route>
        <domain-id>92</domain-id>
        <hop-count>1</hop-count>
        <out-port>129</out-port>
        <in-ports>
            <port>130</port>
            <port>131</port>
            <port>132</port>
            <port>146</port>
            <port>163</port>
            <port>165</port>
            <port>166</port>
            <port>167</port>
            <port>168</port>
            <port>169</port>
            <port>170</port>
            <port>173</port>
            <port>174</port>
            <port>248</port>
            <port>249</port>
            <port>251</port>
        </in-ports>
    </topology-route>
...
    <topology-route>
        <domain-id>239</domain-id>
        <hop-count>2</hop-count>
        <out-port>167</out-port>
        <in-ports>
            <port>128</port>
            <port>129</port>
            <port>130</port>
            <port>131</port>
            <port>146</port>
            <port>159</port>
            <port>160</port>
            <port>163</port>
            <port>165</port>
            <port>166</port>
            <port>168</port>
            <port>169</port>
```

```
            <port>170</port>
            <port>173</port>
            <port>174</port>
            <port>248</port>
            <port>249</port>
            <port>251</port>
        </in-ports>
    </topology-route>
</Response>
```

brocade-fibrechannel-trunk/performance

Description

The performance statistics for each trunk in the switch.

Parameters

Table 66: brocade-fibrechannel-trunk/performance

Name	Type	Description
group	fibrechannel:user-port-number-type	A positive number that uniquely identifies a trunk starting from 1 to the maximum number of trunks in the switch.
tx-bandwidth	uint32	The transmit bandwidth.
tx-capacity	decimal64	The maximum transmit side frame transmission capacity.
tx-utilization	uint64	The transmit side frame transmission rate.
tx-throughput	uint64	The transmit side frame transmission rate. Deprecated: Please use 'tx-utilization' leaf instead.
tx-percentage	fibrechannel:percentage-type	The transmit bandwidth as a percentage of the maximum bandwidth.
rx-bandwidth	uint32	The receive bandwidth.
rx-capacity	decimal64	The maximum receive side frame reception capacity.
rx-utilization	uint64	The receive side frame reception rate.
rx-throughput	uint64	The receive side frame reception rate. Deprecated: Please use 'rx-utilization' leaf instead.
rx-percentage	fibrechannel:percentage-type	The receive bandwidth as a percentage of the maximum bandwidth.
txrx-bandwidth	uint32	The combined (transmit and receive) frame bandwidth.
txrx-capacity	decimal64	The maximum combined (transmit and receive) frame capacity.
txrx-utilization	uint64	The combined (transmit and receive) side frame rate.
txrx-throughput	uint64	The combined (transmit and receive) side frame rate. Deprecated: Please use 'txrx-utilization' leaf instead.
txrx-percentage	fibrechannel:percentage-type	The combined (transmit and receive) bandwidth as a percentage of the maximum bandwidth.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <performance>
  <group>1</group>
  <tx-bandwidth>32</tx-bandwidth>
  <tx-throughput>8000</tx-throughput>
  <tx-percentage>0.00</tx-percentage>
  <rx-bandwidth>32</rx-bandwidth>
  <rx-throughput>0</rx-throughput>
  <rx-percentage>0.00</rx-percentage>
  <txrx-bandwidth>64</txrx-bandwidth>
  <txrx-throughput>8000</txrx-throughput>
  <txrx-percentage>0.00</txrx-percentage>
 </performance>
...
 <performance>
  <group>3</group>
  <tx-bandwidth>32</tx-bandwidth>
  <tx-throughput>0</tx-throughput>
  <tx-percentage>0.00</tx-percentage>
  <rx-bandwidth>32</rx-bandwidth>
  <rx-throughput>8000</rx-throughput>
  <rx-percentage>0.00</rx-percentage>
  <txrx-bandwidth>64</txrx-bandwidth>
  <txrx-throughput>8000</txrx-throughput>
  <txrx-percentage>0.00</txrx-percentage>
 </performance>
</Response>
```

brocade-fibrechannel-trunk/trunk

Description

The list of configured members in the trunk.

Parameters

Table 67: brocade-fibrechannel-trunk/trunk

Name	Type	Description
group	fibrechannel:user-port-number-type	The unique number on the switch for identifying a Fibre Channel trunk.
source-port	fibrechannel:user-port-number-type	The user port number of the trunk member of the local switch.
master	boolean	Indicates whether this link has the trunk controller role. true : This link is the trunk controller. false: This link is a trunk agent.
destination-port	fibrechannel:user-port-number-type	The user port number of the trunk member of the adjacent switch.
neighbor-wwn	fibrechannel:wwn-type	The Fibre Channel WWN of the adjacent switch.
neighbor-switch-name	string	The Fibre Channel user-friendly name of the adjacent switch.
neighbor-domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
deskew	fibrechannel:deskew-type	This parameters determines if ports can be trunked if the variation between different links in a trunk is within range.
trunk-type	fibrechannel:fibrechannel-trunk-type	String enumeration describing the type of trunk.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <trunk>
        <group>1</group>
        <source-port>132</source-port>
        <destination-port>47</destination-port>
        <neighbor-wwn>10:00:d1:1f:cc:1a:f1:1e</neighbor-wwn>
        <neighbor-switch-name>G720</neighbor-switch-name>
        <neighbor-domain-id>238</neighbor-domain-id>
        <deskew>5</deskew>
```

```
        <master>true</master>
        <trunk-type>inter-switch-link</trunk-type>
    </trunk>
...
    <trunk>
        <group>6</group>
        <source-port>251</source-port>
        <destination-port>9</destination-port>
        <neighbor-wwn>10:00:28:24:21:22:2b:d2</neighbor-wwn>
        <neighbor-switch-name>B7810</neighbor-switch-name>
        <neighbor-domain-id>237</neighbor-domain-id>
        <deskew>0</deskew>
        <master>true</master>
        <trunk-type>inter-switch-link</trunk-type>
    </trunk>
</Response>
```

brocade-fibrechannel-trunk/trunk-area

Description

A list of trunk area groups for a switch. You can configure the trunk index for ports in the same octet for F_Ports. Note that this operation is persistent and disruptive.

Parameters

Table 68: brocade-fibrechannel-trunk/trunk-area

Name	Type	Description
trunk-index	fibrechannel:user-port-number-type	The persistent trunk area for range of ports in the F_Port trunk.
master-port	fibrechannel:slot-port-name-type	The trunk controller port
trunk-active	boolean	Whether trunking is active on the port. true : Trunking is active on the port. false: Trunking is not active on the port.
trunk-members	Container	The trunk members of the F_Port trunk group.
- trunk-member	fibrechannel:slot-port-name-type	The individual trunk member of an F_Port trunk in the format slot/port.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <trunk-area>
  <trunk-index>0</trunk-index>
  <trunk-active>true</trunk-active>
  <master-port>0/0</master-port>
  <trunk-members>
   <trunk-member>0/0</trunk-member>
   <trunk-member>0/6</trunk-member>
   <trunk-member>0/7</trunk-member>
  </trunk-members>
 </trunk-area>
 <trunk-area>
  <trunk-index>40</trunk-index>
  <trunk-active>false</trunk-active>
  <master-port/>
  <trunk-members>
   <trunk-member>0/40</trunk-member>
   <trunk-member>0/41</trunk-member>
  </trunk-members>
 </trunk-area>
</Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-ficon/cup

Description

FICON Control Unit Port(CUP) configuration parameters.

Parameters

Table 69: brocade-ficon/cup

Name	Type	Description
fmsmode-enabled	boolean	Indicates whether Ficon Management Server(FMS) Mode is enabled in the logical switch. FMS mode can only be enabled through the rest interface. If a set request is received with fmsmode-enabled=false an error will be returned. false : FMS Mode disabled true : FMS Mode enabled
programmed-offline-state-control	boolean	The Programmed Offline State Control property indicates whether programmed control of the offline state is allowed or prohibited. Programmed control refers to an IBM host sending the 0xFD Set Offline CCW to the CUP to disable a logical switch. false : Programmed control is prohibited true : Programmed control is allowed
user-alert-mode	boolean	The User Alert Mode property indicates whether User Alert Mode is enabled or disabled. When enabled, this mode causes the EFC Server to send a general warning indication to the EFC Server user whenever the EFC Server user attempts an action which will write FICON Director parameters. The EFC Server user may override this warning. When disabled, the general warning indication does not take place. false : User Alert Mode is disabled. true : User Alert Mode is enabled.
active-equal-saved-mode	boolean	The Active=Saved Mode property indicates whether Active=Saved Mode is enabled or disabled. Active=Saved Mode controls updating of the CUP IPL configuration file. When Active=Saved Mode is enabled the IPL file is updated with configuration changes along with the active configuration. When Active=Saved Mode is disabled the IPL file is not updated with changes to the active configuration. In this case a reboot or activation of another configuration file will lose configuration changes made to the active configuration. false : Active=Saved Mode is disabled. true : Active=Saved Mode is enabled.
director-clock-alert-mode	boolean	The Director Clock Alert Mode property indicates whether Director Clock Alert Mode is enabled or disabled. When this mode is enabled a general warning will be displayed on the EFC server when a user attempts to set the Time-Stamp clock. When enabled this mode also causes an error message to be displayed when a user attempts to activate any function that automatically sets the Time-Stamp clock. false : Director Clock Alert Mode is disabled. true : Director Clock Alert Mode is enabled.

Name	Type	Description
mihpto	uint32	Contains the current value of the Missing Interrupt Handler Primary Timeout. This value is only available when FMS mode is enabled. This value cannot be changed through the rest api. See the IBM PI Spec for definitions of the contents.
unsolicited-alert-mode-fru-enabled	boolean	Indicates if Unsolicited Alerts for FRU events are enabled. false : FRU alerts disabled true : FRU alerts enabled
unsolicited-alert-mode-hsc-enabled	boolean	Indicates if Unsolicited Alerts for Health Summary Code(HSC) events are enabled. false : HSC alerts disabled true : HSC alerts enabled
unsolicited-alert-mode-invalid-attach-enabled	boolean	Indicates if Unsolicited Alerts for Invalid Attachment events are enabled. false : Invalid Attachment alerts disabled true : Invalid Attachment alerts enabled

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <cup>
  <fmsmode-enabled>true</fmsmode-enabled>
  <programmed-offline-state-control>true</programmed-offline-state-control>
  <user-alert-mode>false</user-alert-mode>
  <active-equal-saved-mode>true</active-equal-saved-mode>
  <director-clock-alert-mode>true</director-clock-alert-mode>
  <mihpto>180</mihpto>
  <unsolicited-alert-mode-fru-enabled>true</unsolicited-alert-mode-fru-enabled>
  <unsolicited-alert-mode-hsc-enabled>true</unsolicited-alert-mode-hsc-enabled>
  <unsolicited-alert-mode-invalid-attach-enabled>true</unsolicited-alert-mode-invalid-attach-enabled>
 </cup>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-ficon/lirr

Description

A list of Link Incident Registration Record(LIRR) records from the FICON database.

All LIRR records for the entire fabric are returned unless a link-address is specified to return only a single LIRR record.

The LIRR records are simply reporting the state data for registered devices so you cannot create or update LIRR records.

Parameters

Table 70: brocade-ficon/lirr

Name	Type	Description
link-address	fibrechannel:brocade-hex-string-type	The link address(first two bytes of Port Identifier(PID)) of the LIRR record.
format	fibrechannel:brocade-hex-string-type	The Format field from the LIRR record. 0x18 : indicates the record is a FICON node identifier
port-type	fibrechannel:port-type-string-type	The port type of the LIRR record. i.e. n-port, e-port, etc. See fibrechannel:port-type-string-type for specific values
listener-port-wwn	fibrechannel:wwn-type	The Listener Port WWN
switch-port-wwn	fibrechannel:wwn-type	The Switch Port WWN
listener-type	string	The Listener Type field from the LIRR record. For example, Conditional-C, Conditional-S, etc.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
  <?xml version="1.0"?>
 <Response>
  <lirr>
   <format>0x18</format>
   <port-type>n-port</port-type>
```

```
   <link-address>0x0128</link-address>
   <listener-port-wwn>c0:20:22:d2:2f:02:11:21</listener-port-wwn>
   <switch-port-wwn>20:22:c2:f2:2c:01:2a:21</switch-port-wwn>
   <listener-type>Conditional-C</listener-type>
  </lirr>
 ...
  <lirr>
   <format>0x18</format>
   <port-type>n-port</port-type>
   <link-address>0x012b</link-address>
   <listener-port-wwn>c0:20:22:d2:2f:02:11:21</listener-port-wwn>
   <switch-port-wwn>20:22:c2:f2:2c:01:2a:21</switch-port-wwn>
   <listener-type>Conditional</listener-type>
  </lirr>
 </Response>
```

brocade-ficon/logical-path

Description

A list of Logical Paths that are currently established to the Control Unit Port(CUP).

Setting the Current Reporting Path is supported through this interface. To set the current reporting path you must supply the key(link-address and channel-image-id) and a reporting-path-state value of Current.

Parameters

Table 71: brocade-ficon/logical-path

Name	Type	Description
link-address	fibrechannel:brocade-hex-string-type	The link address(first two bytes of Port Identifier(PID)) of the logical path. This is part of the composite key that must be supplied to set the Current Reporting Path.
channel-image-id	fibrechannel:brocade-hex-string-type	The channel Image ID of the logical path. This is part of the composite key that must be supplied to set the Current Reporting Path.
operational-state	logical-path-operational-states	The operational state of the logical path.
reporting-path-state	string	The reporting path state of the logical path. The reporting path state could be zero or more of the following values Primary : Primary Reporting Path Alternate : Alternate Reporting Path Current : Current Reporting Path To set the Current Reporting Path the reporting-path-state property must be supplied and the only valid value that can be supplied is Current.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.


```
"Response": {
 "logical-path": [
  {
   "link-address": "0x012b",
   "channel-image-id": "0x03",
   "operational-state": "Oper",
   "reporting-path-state": "Alternate"
  },
  {
   "link-address": "0x012a",
   "channel-image-id": "0x03",
   "operational-state": "Oper",
   "reporting-path-state": "Current"
  },
  {
   "link-address": "0x0129",
   "channel-image-id": "0x03",
   "operational-state": "Reset",
   "reporting-path-state": ""
  },
  {
   "link-address": "0x0128",
   "channel-image-id": "0x03",
   "operational-state": "Oper",
   "reporting-path-state": ""
  }
  ]
 }
}
```


PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-ficon/rlir

Description

A list of Registered Link Incident Record(RLIR) records from the FICON database.

All RLIR records for the entire fabric. There is no unique key to allow selection of a single RLIR record.

The RLIR records are simply reporting the state data for connected devices so you cannot create or update RLIR records.

Some of the properties correspond to the definition of the RLIR record in FC-LS and the RNID record in FC-SB and additional definition can be found in the specs.

Parameters

Table 72: brocade-ficon/rlir

Name	Type	Description
link-address	fibrechannel:brocade-hex-string-type	The link address(first two bytes of Port Identifier(PID)) of the RLIR record.
format	fibrechannel:brocade-hex-string-type	The Format field from the RLIR record. 0x18 : indicates the record is a FICON node identifier
port-type	fibrechannel:port-type-string-type	The port type of the RLIR record. i.e. n-port, e-port, etc. See fibrechannel:port-type-string-type for specific values
port-index	fibrechannel:user-port-number-type	The Physical Port Number field from the RLIR record.
incident-count	uint8	The Incident Count field from the RLIR record. This number increases by 1 for each incident within the switch.
time-stamp-format	uint8	The Time Stamp Format field from the RLIR record.
time-stamp	string	An ASCII string representation of the Time Stamp field from the RLIR record.
port-status	string	The Port Status field from the RLIR record. This value is returned as a string indicating the status. i.e. Link not operational
link-failure-type	string	The Link Failure Type field from the RLIR record. This value is returned as a string indicating the type of link failure. i.e. Loss of Signal or synchronization
registered-port-wwn	fibrechannel:wwn-type	The Registered Port WWN
registered-node-wwn	fibrechannel:wwn-type	The Registered Node WWN
port-flags	fibrechannel:brocade-hex-string-type	The Flags field from the the RLIR record.

Name	Type	Description
port-node-parameters	fibrechannel:brocade-hex-string-type	The Node Parameters field from the RLIR record.
port-type-number	string	The Type Number of the node that describes the machine type of the device.
port-model-number	string	The Model Number of the device
port-manufacturer	string	The Model Number or code of the device
port-plant	string	The Manufacturer Plant name or code of the device
port-sequence-number	string	The Sequence Number of the device
port-tag	fibrechannel:brocade-hex-string-type	The physical identifier of the device
switch-port-wwn	fibrechannel:wwn-type	The Switch Port WWN
switch-node-wwn	fibrechannel:wwn-type	The Switch Node WWN
switch-flags	fibrechannel:brocade-hex-string-type	The Flags field from the the RLIR record.
switch-node-parameters	fibrechannel:brocade-hex-string-type	The Node Parameters field from the RLIR record.
switch-type-number	string	The Type Number of the node that describes the machine type of the device.
switch-model-number	string	The Model Number of the device
switch-manufacturer	string	The Model Number or code of the device
switch-plant	string	The Manufacturer Plant name or code of the device
switch-sequence-number	string	The Sequence Number of the device
switch-tag	fibrechannel:brocade-hex-string-type	The physical identifier of the device

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <rlir>
  <format>0x18</format>
  <port-type>f-port</port-type>
  <link-address>0x012b</link-address>
```

```
   <port-index>43</port-index>
   <incident-count>1</incident-count>
   <time-stamp-format>1</time-stamp-format>
   <time-stamp>Tue May 21 16:42:46 2019</time-stamp>
   <port-status>Link not operational</port-status>
   <link-failure-type>Not operational seq. recognized</link-failure-type>
   <registered-port-wwn>c0:50:76:d5:5f:08:11:d1</registered-port-wwn>
   <registered-node-wwn>50:05:07:64:00:cf:c8:87</registered-node-wwn>
   <port-flags>0x10</port-flags>
   <port-node-parameters>0x31014f</port-node-parameters>
   <port-type-number>003906</port-type-number>
   <port-model-number>M02</port-model-number>
   <port-manufacturer>IBM</port-manufacturer>
   <port-plant>02</port-plant>
   <port-sequence-number>0000000FC887</port-sequence-number>
   <port-tag>0xc04f</port-tag>
   <switch-port-wwn>20:2b:c4:f5:7c:01:2a:21</switch-port-wwn>
   <switch-node-wwn>10:00:c4:f5:7c:01:2a:21</switch-node-wwn>
   <switch-flags>0x00</switch-flags>
   <switch-node-parameters>0x200a2b</switch-node-parameters>
   <switch-type-number>SLKWRM</switch-type-number>
   <switch-model-number>630</switch-model-number>
   <switch-manufacturer>BRD</switch-manufacturer>
   <switch-plant>CA</switch-plant>
   <switch-sequence-number>1FKF0324N005</switch-sequence-number>
   <switch-tag>0x002b</switch-tag>
  </rlir>
 </Response>
```

brocade-ficon/rnid

Description

A list of Read Node Identifier(RNID) records from the FICON database.
All RNID records for the entire fabric are returned unless a
link-address is specified to return only a single RNID record.
The RNID records are simply reporting the state data for connected
devices so you cannot create or update RNID records.

Most of the properties correspond to the definition of the RNID record
in FC-SB and additional definition can be found in the spec.

Parameters

Table 73: brocade-ficon/rnid

Name	Type	Description
link-address	fibrechannel:brocade-hex-string-type	The link address(first two bytes of Port Identifier(PID)) of the RNID record.
format	fibrechannel:brocade-hex-string-type	The Format field from the RNID record. 0x18 : indicates the record is a FICON node identifier
port-type	fibrechannel:port-type-string-type	The port type of the RNID record. i.e. n-port, e-port, etc. See the fibrechannel:port-type-string-type for more information.
registered-port-wwn	fibrechannel:wwn-type	The Registered Port WWN
registered-node-wwn	fibrechannel:wwn-type	The Registered Node WWN
flags	fibrechannel:brocade-hex-string-type	The Flags field from the RNID record.
node-parameters	fibrechannel:brocade-hex-string-type	The Node Parameters field from the RNID record.
type-number	string	The Type Number of the node that describes the machine type of the device
model-number	string	The Model Number of the device
manufacturer	string	The Model Number or code of the device
plant	string	The Manufacturer Plant name or code of the device
sequence-number	string	The Sequence Number of the device
tag	fibrechannel:brocade-hex-string-type	The physical identifier of the device

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <rnid>
  <format>0x18</format>
  <port-type>n-port</port-type>
  <link-address>0x0128</link-address>
  <registered-port-wwn>c0:20:26:d2:2f:02:11:d1</registered-port-wwn>
```

```
   <registered-node-wwn>50:02:02:24:00:cf:c2:27</registered-node-wwn>
   <flags>0x10</flags>
   <node-parameters>0x31014c</node-parameters>
   <type-number>003906</type-number>
   <model-number>M02</model-number>
   <manufacturer>IBM</manufacturer>
   <plant>02</plant>
   <sequence-number>0000000FC887</sequence-number>
   <tag>0xc04c</tag>
  </rnid>
 ...
  <rnid>
   <format>0x18</format>
   <port-type>n-port</port-type>
   <link-address>0x012b</link-address>
   <registered-port-wwn>c0:20:26:d2:2f:02:11:d1</registered-port-wwn>
   <registered-node-wwn>50:02:02:24:00:cf:c2:27</registered-node-wwn>
   <flags>0x10</flags>
   <node-parameters>0x31014f</node-parameters>
   <type-number>003906</type-number>
   <model-number>M02</model-number>
   <manufacturer>IBM</manufacturer>
   <plant>02</plant>
   <sequence-number>0000000FC887</sequence-number>
   <tag>0xc04f</tag>
  </rnid>
 </Response>
```

brocade-ficon/switch-rnid

Description

A list of Switch Read Node Identifier(RNID) records from the FICON database.

All Switch RNID records for the entire fabric are returned unless a switch-addr is specified to return only a single Switch RNID record.

The Switch RNID records are simply reporting the state data for connected devices so you cannot create or update Switch RNID records.

Most of the properties correspond to the definition of the RNID record in FC-SB and additional definition can be found in the spec.

Parameters

Table 74: brocade-ficon/switch-rnid

Name	Type	Description
switch-address	fibrechannel:brocade-hex-string-type	The switch address(domain ID) of the Switch RNID record.
switch-wwn	fibrechannel:wwn-type	The Switch Node WWN
flags	fibrechannel:brocade-hex-string-type	The Flags field from the Switch RNID record.
node-parameters	fibrechannel:brocade-hex-string-type	The Node Parameters field from the Switch RNID record.
type-number	string	The Type Number of the node that describes the machine type of the switch.
model-number	string	The Model Number of the switch
manufacturer	string	The Model Number or code of the switch
plant	string	The Manufacturer Plant name or code of the switch
sequence-number	string	The Sequence Number of the switch
tag	fibrechannel:brocade-hex-string-type	The physical identifier of the switch

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <switch-rnid>
  <switch-address>0x01</switch-address>
  <switch-wwn>10:00:c4:f5:7c:01:2a:21</switch-wwn>
  <flags>0x00</flags>
  <node-parameters>0x200a00</node-parameters>
  <type-number>SLKWRM</type-number>
  <model-number>630</model-number>
  <manufacturer>BRD</manufacturer>
  <plant>CA</plant>
  <sequence-number>1FKF0324N005</sequence-number>
  <tag>0x01ff</tag>
```

```
  </switch-rnid>
</Response>
```

brocade-firmware/firmware-config

Description

The firmware check switch-level config and operational parameters.

This is used to check the integrity of software image that is stored

on a switch. Obsolete because firmwarecheck is deprecated.

Parameters

Table 75: brocade-firmware/firmware-config

Name	Type	Description
firmware-check-enabled-state	boolean	The firmwarecheck state of the switch. The possible values are: true : Enables firmware integrity check on every reboot. false : Disables firmware integrity check on every reboot.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <firmware-config>
        <firmware-check-enabled-state>false</firmware-check-enabled-state>
    </firmware-config>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-firmware/firmware-history

Description

The firmware versions and the download history.

For each firmware download record, displays the date and time,

the switch name, the slot number, the process ID, and the

firmware version.

Parameters

Table 76: brocade-firmware/firmware-history

Name	Type	Description
sequence-number	uint16	The sequence number in the order in which the firmwares were installed on the switch.
time-stamp	yang:date-and-time	The date and time when the firmware was installed on the switch.
switch-name	string	A printable ASCII name for the switch.
slot-number	uint16	The slot number.
process-id	uint32	The process ID which installed the firmware on the switch.
firmware-version	string	A human readable string identifying the firmware version running on the switch.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <firmware-history>
  <sequence-number>1</sequence-number>
  <time-stamp>Mon Mar 1 21:11:55 2021</time-stamp>
  <switch-name>unknown(0)</switch-name>
  <slot-number>2</slot-number>
  <process-id>2063</process-id>
  <firmware-version>Fabos Version 9.1.0fos_major_210226_015811</firmware-version>
```

```
  </firmware-history>
...
  <firmware-history>
   <sequence-number>5</sequence-number>
   <time-stamp>Thu Mar 4 09:26:00 2021</time-stamp>
   <switch-name>V21SW</switch-name>
   <slot-number>2</slot-number>
   <process-id>2294</process-id>
   <firmware-version>Fabos Version 9.1.0FWHlatest</firmware-version>
  </firmware-history>
</Response>
```

brocade-fru/blade

Description

A list of blade details for the specified slot number.

Parameters

Table 77: brocade-fru/blade

Name	Type	Description
time-alive	fru:time-detail-type	The number of days the FRU has been powered on.
time-awake	fru:time-detail-type	The number of days since the FRU was last powered on.
part-number	fru:part-number-type	The part number assigned by the organization responsible for producing or manufacturing the blade.
serial-number	fru:serial-number-type	The serial-number resource contains a printable ASCII string that specifies the serial number of the chassis.
vendor-serial-number	string	The serial number of the chassis assigned by the vendor.
vendor-part-number	string	The part number of the chassis assigned by the vendor.
vendor-revision-number	string	The revision number of the chassis assigned by the vendor.
vendor-id	string	The identifier of the chassis assigned by the vendor.
slot-number	uint16	The number of the physical slot in the chassis in which the blade is inserted.
manufacturer	string	The name of the blade manufacturer.
blade-type	blade-type-type	The type of blade (sw blade, cp blade, core blade, ap blade).
blade-id	uint16	The ID of the blade.
blade-state	blade-state-type	The current state of the blade.
model-name	fibrechannel:product-name-type	A printable ASCII string that describes the model of the blade.
firmware-version	fru:firmware-version-type	A human-readable string that displays the firmware version running on the switch. For an AP blade, it displays the AP blade firmware version. For other blades, it displays the active CP firmware version.
fc-port-count	uint16	The number of FC ports supported by the blade.
ge-port-count	uint16	The number of GE ports supported by the blade.

Name	Type	Description
link-local-address	union {inet:ipv6-address;inet:ipv6-prefix;}	IPv6 Link local address of CP blade. This parameter is applicable only for CP blades.
subnet-mask	inet:ipv4-address	The IPv4 subnet mask of the switch IP network. This parameter is applicable only for CP blades. Deprecated: Please use 'static-subnet-mask' and 'dhcp-subnet-mask' in brocade-fru instead.
primary-firmware-version	fru:firmware-version-type	A human-readable string that identifies the firmware version running on the primary partition of the CP blade.
secondary-firmware-version	fru:firmware-version-type	A human readable string that identifies the firmware version running on the secondary partition of the CP blade.
power-consumption	fru:power-detail-type	The maximum power consumption allocated for the blade.
power-usage	fru:power-detail-type	The real-time power consumed by the FRU.
extension-enabled	boolean	Indicates whether the switch or blade supports extension. For blade ID's 154, 186, and 213 the value is set to true. false : Extension is not supported. true : Extension is supported.
extension-app-mode	string	The application mode configuration of the extension blade or switch. Note that the application mode configuration change is a disruptive operation and that when the configuration change is successful, one of the following actions occur: switch : A switch automatically reboots. blade : A blade automatically powers off and on. Blade ID 213 supports only hybrid mode. For blade ID 154 and 186, the default is FCIP mode. For blade ID 213, the default is hybrid mode. Supported modes: FCIP : Supports FCIP-only tunnels. hybrid : Supports FCIP with IPEXT tunnels. unavailable : The value is unavailable at the moment. This is read-only value and can be seen in specific scenarios where the information is not loaded because of slot power off or slot faulty after reboot or powercycle.

Name	Type	Description
extension-ve-mode	string	The VE mode configuration of the extension blade or switch. Note that a VE mode configuration change is a disruptive operation and that when the configuration change is successful, one of the following actions occur: switch : A switch automatically reboots. blade : A blade automatically powers off and on. For blade ID 154 and 186, the default is 10VE mode. For blade ID 213, the default is not applicable. Supported modes: not applicable : Not a supported configuration. 10VE : 10 VEs available for tunnel configuration (5 VE_Ports per DP). 20VE : 20 VEs available for tunnel configuration (10 VE_Ports per DP). unavailable : The value is unavailable at the moment. This is a read-only value and can be seen in specific scenarios where the information is not loaded because of slot power off or slot faulty after reboot or powercycle.
extension-ge-mode	string	The GE mode configuration of the extension blade or switch. In copper mode, the RJ-45 ports (ge0 and ge1) are enabled and optical ports (ge2 and ge3) are disabled. In optical mode, the optical ports (ge2 and ge3) are enabled and the RJ45 ports (ge0 and ge1) are disabled. For blade ID 154 and 186, the default is not applicable. For blade ID 213, the default is optical. Supported modes: not applicable : Not a supported configuration. optical : ge2 and ge3 use the first two optical ports. copper : ge0 and ge1 use front end RJ45 ports. unavailable : The value is unavailable at the moment. This is read-only value and can be seen in specific scenarios where the information is not loaded because of slot power off or slot faulty after reboot or powercycle.
manufacture-date	fru:fru-date-type	The FRU manufacture date.
last-header-update-date	fru:fru-date-type	The date since the last FRU header was updated.
header-version	uint16	The FRU header version number.
hardware-version	fibrechannel:brocade-hex-string-type	The FRU Hardware version.
ip-address-list	Container	A list of IP addresses assigned to the CP blade. Deprecated: Please use 'static-ip-addresses' and 'stateful-ip-addresses' leafs in brocade-fru instead.
- ip-address	union {inet:ip-address;inet:ip-prefix;}	List of IP addresses assigned to the CP blade. This parameter is applicable only for CP blades.
static-ip-addresses	Container	A list of out-of-band static IPv4 and IPv6 addresses assigned to the ethernet port of CP blade.
- ip-address	inet:ip-address	List of static IP addresses assigned to the ethernet port of CP blade. This parameter is applicable only for CP blades.

Name	Type	Description
stateful-ip-addresses	Container	List of out-of-band DHCP and DHCPv6 addresses assigned to the ethernet port of CP blade.
- ip-address	union {inet:ip-address;inet:ip-prefix;}	List of out-of-band stateful IP addresses assigned to the ethernet port of CP blade. This parameter is applicable only for CP blades.
stateless-ipv6-addresses	Container	List of out-of-band stateless IPv6 addresses assigned to the ethernet port of CP blade.
- ip-address	union {inet:ipv6-address;inet:ipv6-prefix;}	List of out-of-band stateless IP addresses assigned to the ethernet port of CP blade. This parameter is applicable only for CP blades.
ip-gateway-list	Container	A list of gateway IP address of the CP blade. Deprecated: Please use 'static-ip-gateways', 'dhcp-gateway' and 'stateless-ip-gateways' leafs in brocade-management-ip-interface instead.
- ip-gateway	fru:ip-gateway-type	The list of gateway IP addresses of an IP router that can route packets to the destination IP address. The gateway address must be on the same IP subnet as one of the port IP addresses. This parameter is applicable only for CP blades.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.


```
{
    "Response": {
        "blade": [
            {
                "slot-number": 1,
                "part-number": "60-1003201-12",
                "serial-number": "DYK3213M01R",
                "vendor-part-number": "test",
                "vendor-serial-number": "infodot-252267",
                "vendor-revision-number": "23",
                "vendor-id": "BROCADE",
                "manufacturer": "Brocade Communications Systems LLC",
                "blade-type": "cp blade",
                "firmware-version": "v9.2.0",
                "primary-firmware-version": "v9.2.0",
                "secondary-firmware-version": "v9.2.0",
```

```
                "ip-address-list": {
                    "ip-address": [
                        "10.10.10.41",
                        "2020:100:0:fa00:c0f0:0cff:fe10:a0d1/64"
                    ]
                },
                "static-ip-addresses": {
                    "ip-address": [
                        "10.10.10.41"
                    ]
                },
                "stateful-ip-addresses": {},
                "stateless-ipv6-addresses": {
                    "ip-address": [
                        "2020:100:0:fa00:c0f0:0cff:fe10:a0d1/64"
                    ]
                },
                "link-local-address": "fe10::c1f1:1cff:fe11:a1d1/64",
                "subnet-mask": "255.255.225.0",
                "ip-gateway-list": {
                    "ip-gateway": [
                        "10.10.10.1",
                        "fe10::126"
                    ]
                },
                "model-name": "CPX6",
                "extension-enabled": false,
                "blade-id": 175,
                "blade-state": "enabled",
                "power-usage": -48,
                "time-alive": 1556,
                "time-awake": 0,
                "power-consumption": -50,
                "manufacture-date": "Day:  4  Month:  4  Year: 2016",
                "last-header-update-date": "Day: 28  Month:  2  Year: 2023",
                "header-version": 2,
                "hardware-version": "0x5"
            },
...

            {
                "slot-number": 8,
                "part-number": "60-1003847-01",
                "serial-number": "FDU0325N021",
                "vendor-serial-number": "infodot-232069",
                "vendor-id": "",
                "manufacturer": "Brocade Communications Systems LLC",
                "blade-type": "sw blade",
                "firmware-version": "v9.2.0",
                "model-name": "FC32-64",
                "fc-port-count": 64,
                "extension-enabled": false,
                "blade-id": 204,
```

```
                "blade-state": "enabled",
                "power-usage": -194,
                "time-alive": 1493,
                "time-awake": 0,
                "power-consumption": -387,
                "manufacture-date": "Day: 23  Month:  6  Year: 2017",
                "last-header-update-date": "Day: 28  Month:  2  Year: 2023",
                "header-version": 2,
                "hardware-version": "0x2"
            }
        ]
    }
}
```


PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-fru/fan

Description

The details about the fan units.

Parameters

Table 78: brocade-fru/fan

Name	Type	Description
time-alive	fru:time-detail-type	The number of days the FRU has been powered on.
time-awake	fru:time-detail-type	The number of days since the FRU was last powered on.
part-number	fru:part-number-type	The part number assigned by the organization responsible for producing or manufacturing the blade.
serial-number	fru:serial-number-type	The serial-number resource contains a printable ASCII string that specifies the serial number of the chassis.
vendor-serial-number	string	The serial number of the chassis assigned by the vendor.
vendor-part-number	string	The part number of the chassis assigned by the vendor.
vendor-revision-number	string	The revision number of the chassis assigned by the vendor.
vendor-id	string	The identifier of the chassis assigned by the vendor.
unit-number	uint16	The physical slot number in the chassis where the fan is located.
power-consumption	fru:power-detail-type	The maximum power consumption allocated for the fan.
operational-state	string	The current operational state of the Fan. The state 'not ok' displays when the unit is faulty for an unknown reason.
speed	uint32	The fan speed.
airflow-direction	air-flow-direction-type	The air flow direction of the fan blowers.
manufacture-date	fru:fru-date-type	The FRU manufacture date.

Name	Type	Description
last-header-update-date	fru:fru-date-type	The date since the last FRU header was updated.
header-version	uint16	The FRU header version number.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <fan>
  <unit-number>1</unit-number>
  <serial-number>DYL3017M0BG</serial-number>
  <part-number>60-1003203-04</part-number>
  <power-consumption>-300</power-consumption>
  <operational-state>ok</operational-state>
  <airflow-direction>non-portside intake</airflow-direction>
  <speed>5076</speed>
  <time-alive>435</time-alive>
  <time-awake>2</time-awake>
 </fan>
 <fan>
  <unit-number>2</unit-number>
  <serial-number>DYL3017M06K</serial-number>
  <part-number>60-1003203-04</part-number>
  <power-consumption>-300</power-consumption>
  <operational-state>ok</operational-state>
  <airflow-direction>non-portside intake</airflow-direction>
  <speed>5012</speed>
  <time-alive>435</time-alive>
  <time-awake>2</time-awake>
 </fan>
</Response>
```

brocade-fru/history-log

Description

The entire history log records of all the FRU's.

Parameters

Table 79: brocade-fru/history-log

Name	Type	Description
part-number	fru:part-number-type	The part number assigned by the organization responsible for producing or manufacturing the blade.
serial-number	fru:serial-number-type	The serial-number resource contains a printable ASCII string that specifies the serial number of the chassis.
fru-type	fru:fru-type	The type of the fru.
position	uint16	The physical location where the fru is located. For fan and powersupply it is unit number. For slot it is slot number.
state	fru:state-type	The state of the fru.
time-stamp	yang:date-and-time	Timestamp of the event generated.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <history-log>
  <fru-type>core blade</fru-type>
  <position>5</position>
  <state>inserted</state>
  <time-stamp>2017-06-14T13:11:31-06:00</time-stamp>
  <serial-number>DZD3213M011</serial-number>
  <part-number>60-1003226-10</part-number>
 </history-log>
...
 <history-log>
  <fru-type>sw blade</fru-type>
  <position>8</position>
  <state>inserted</state>
  <time-stamp>2019-01-04T10:53:43-07:00</time-stamp>
  <serial-number>FDU0325N021</serial-number>
  <part-number>60-1003847-01</part-number>
 </history-log>
```

```
</Response>
```

brocade-fru/power-supply

Description

The details about the power supply units.

Parameters

Table 80: brocade-fru/power-supply

Name	Type	Description
time-alive	fru:time-detail-type	The number of days the FRU has been powered on.
time-awake	fru:time-detail-type	The number of days since the FRU was last powered on.
part-number	fru:part-number-type	The part number assigned by the organization responsible for producing or manufacturing the blade.
serial-number	fru:serial-number-type	The serial-number resource contains a printable ASCII string that specifies the serial number of the chassis.
vendor-serial-number	string	The serial number of the chassis assigned by the vendor.
vendor-part-number	string	The part number of the chassis assigned by the vendor.
vendor-revision-number	string	The revision number of the chassis assigned by the vendor.
vendor-id	string	The identifier of the chassis assigned by the vendor.
unit-number	uint16	The physical slot number of the chassis where the power supply is located.
power-production	uint32	The maximum power production for the power supply.
input-voltage	decimal64	The input voltage of the power supply unit. Value -1 indicates 'Not Available'.
airflow-direction	air-flow-direction-type	The airflow direction of the power supply fans.
power-source	string	The power supply input voltage type. HVDC : high-voltage direct current HVAC : high-voltage alternating current
operational-state	string	The operational state of the power supply unit.
temperature	decimal64	The temperature of the power supply sensor.
power-usage	fru:power-detail-type	The real-time power consumed by the FRU.
temperature-sensor-supported	boolean	Indicates whether the power supply sensor supports temperature monitoring. false : Temperature monitoring is not supported. true : Temperature monitoring is supported.
manufacture-date	fru:fru-date-type	The FRU manufacture date. The field will be empty if the powersupply is absent.
last-header-update-date	fru:fru-date-type	The date since the last FRU header was updated.
header-version	uint16	The FRU header version number.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "power-supply": [
            {
                "unit-number": 1,
                "serial-number": "DUC2M32M2FD",
                "part-number": "23-0000161-01",
                "power-production": 2870,
                "power-source": "AC",
                "operational-state": "ok",
                "airflow-direction": "non-portside intake",
                "input-voltage": 205.50,
                "temperature": 29.00,
                "time-alive": 1490,
                "time-awake": 0,
                "temperature-sensor-supported": true,
                "vendor-revision-number": "",
                "vendor-part-number": "",
                "vendor-serial-number": "",
                "vendor-id": "",
                "manufacture-date": "Day: 31  Month:  7  Year: 2016",
                "last-header-update-date": "Day: 28  Month:  2  Year: 2023",
                "header-version": 2
            },
            {
                "unit-number": 2,
                "serial-number": "DUC2M32M2ME",
                "part-number": "23-0000161-01",
                "power-production": 2870,
                "power-source": "AC",
                "operational-state": "ok",
                "airflow-direction": "non-portside intake",
                "input-voltage": 205.00,
                "temperature": 28.00,
                "time-alive": 1552,
                "time-awake": 0,
                "temperature-sensor-supported": true,
                "vendor-revision-number": "",
                "vendor-part-number": "",
```

```
                "vendor-serial-number": "",
                "vendor-id": "",
                "manufacture-date": "Day:  1  Month:  8  Year: 2016",
                "last-header-update-date": "Day: 28  Month:  2  Year: 2023",
                "header-version": 2
            }
        ]
    }
}
```

brocade-fru/sensor

Description

The details about sensors.

Parameters

Table 81: brocade-fru/sensor

Name	Type	Description
id	uint16	The number of the sensor id in the chassis.
slot-number	uint16	The number of the physical slot in the chassis in which the blade is inserted.
index	uint16	The sensor index in the specific fru.
state	fru:operational-state-type	The current operational state of the sensor.
category	fru:sensor-category	The type of the sensor.
temperature	uint16	The temperature of the fru sensor. Value '0' indicates 'Not Available'.
temperature-in-fahrenheit	int32	The temperature of the fru sensor in fahrenheit. Value '0' indicates 'Not Available'.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "sensor": [
```

```
            {
                "id": 1,
                "slot-number": 3,
                "index": 0,
                "state": "absent"
            },
            {
                "id": 2,
                "slot-number": 4,
                "index": 0,
                "state": "absent"
            },
...
            {
                "id": 25,
                "slot-number": 8,
                "index": 4,
                "state": "ok",
                "category": "temperature",
                "temperature": 55,
                "temperature-in-fahrenheit": 131
            }
        ]
    }
}
```


brocade-fru/wwn

Description

The details about the wwn card units.

Parameters

Table 82: brocade-fru/wwn

Name	Type	Description
part-number	fru:part-number-type	The part number assigned by the organization responsible for producing or manufacturing the blade.
serial-number	fru:serial-number-type	The serial-number resource contains a printable ASCII string that specifies the serial number of the chassis.
time-alive	fru:time-detail-type	The number of days the FRU has been powered on.
time-awake	fru:time-detail-type	The number of days since the FRU was last powered on.
vendor-serial-number	string	The serial number of the chassis assigned by the vendor.
vendor-part-number	string	The part number of the chassis assigned by the vendor.
vendor-revision-number	string	The revision number of the chassis assigned by the vendor.
vendor-id	string	The identifier of the chassis assigned by the vendor.

Name	Type	Description
unit-number	uint16	The physical slot number of the chassis where the wwn card is located.
operational-state	fru:operational-state-type	The current operational state of the WWN card.
airflow-direction	air-flow-direction-type	The system air flow direction.
power-usage	fru:power-detail-type	The real-time power consumed by the FRU. This parameter is applicable only for chassis. Default value will be displayed if the WWN card is absent
generation-number	uint16	The wwn card generation number. Default value will be displayed if the WWN card is absent
manufacture-date	fru:fru-date-type	The FRU manufacture date. Default value will be displayed if the WWN card is absent
last-header-update-date	fru:fru-date-type	The date since the last FRU header was updated. Default value will be displayed if the WWN card is absent
header-version	uint16	The FRU header version number. Default value will be displayed if the WWN card is absent
time-alive-in-hours	fru:time-detail-type-in-hours	The time elapsed in hours since the FRU has been powered on. Default value will be displayed if the WWN card is absent
time-awake-in-hours	fru:time-detail-type-in-hours	The time elapsed in hours since the FRU was manufactured. Default value will be displayed if the WWN card is absent

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "wwn": [
            {
                "unit-number": 1,
                "serial-number": "DZH3010M018",
                "part-number": "60-1003194-02",
                "generation-number": 6,
                "vendor-revision-number": "25",
                "vendor-part-number": "SLKWRM0000X4S",
                "vendor-serial-number": "infodot-212618",
                "vendor-id": "BRD0000CA",
```

```
                "airflow-direction": "non-portside intake",
                "power-usage": -1,
                "time-alive-in-hours": 39826,
                "time-awake-in-hours": 60136,
                "manufacture-date": "Day: 21  Month:  4  Year: 16",
                "last-header-update-date": "Day:  1  Month:  3  Year: 2023",
                "header-version": 2,
                "operational-state": "ok"
            },
            {
                "unit-number": 2,
                "serial-number": "DZH3010M037",
                "part-number": "60-1003194-02",
                "generation-number": 6,
                "vendor-revision-number": "a",
                "vendor-part-number": "dfasdf",
                "vendor-serial-number": "adfaasdfasdf",
                "vendor-id": "asdf",
                "airflow-direction": "non-portside intake",
                "power-usage": -1,
                "time-alive-in-hours": 39826,
                "time-awake-in-hours": 60136,
                "manufacture-date": "Day: 21  Month:  4  Year: 16",
                "last-header-update-date": "Day:  1  Month:  3  Year: 2023",
                "header-version": 2,
                "operational-state": "ok"
            }
        ]
    }
}
```


brocade-interface/extension-ip-interface

Description

Represents the IP interface defined on extension blade or system.

Parameters

Table 83: brocade-interface/extension-ip-interface

Name	Type	Description
name	brocade-interface-types:ip-extension-interface-type	The name of the interface.
dp-id	uint16	Extension Data Path Processor ID associated with the IP interface. Based on platform either it will have a single DP or dual DP. In case of single DP only DP0 is supported, and in case of dual DP both DP0 and DP1 are supported 0 : DP0 1 : DP1.

Name	Type	Description
ip-address	union {inet:ip-address;inet:ip-prefix;}	Specifies the source IPv4/IPv6 address of the interface.
ip-prefix-length	brocade-interface-types:ip-prefix-length	The prefix length operator for the IP address. Once set, prefix length cannot be changed.
mtu-size	uint16	Specifies the maximum transmission unit size. The permitted range is 1280 Bytes to 9216 Bytes or the special value 1 which is equivalent to 'auto' mode to enable PMTU discovery. The discovered value of MTU in 'auto' mode can be found as a circuit attribute.
vlan-id	uint16	Specifies the VLAN ID. This operand is optional.
status-flags	string	IP interface Flags: U = Up B = Broadcast D = Debug L = Loopback P = Point2Point R = Running I = InUse N = NoArp PR= Promiscous M = Multicast S = StaticArp LU= LinkUp

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message and a “200 OK” status in the headers.

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-interface/fibrechannel

Description

A list of interfaces on the device.

System-controlled interfaces created by the system are
always present in this list, whether they are configured or not.

Parameters

Table 84: brocade-interface/fibrechannel

Name	Type	Description
name	fibrechannel:slot-port-name-type	The name of the interface. The slot and port number of the specified port in the format slot/port.
wwn	fibrechannel:wwn-type	The world wide name (WWN) of the specified port.
operational-status	uint32	Deprecated. Please use operational-status-string in stead. The current operational status of the specified port. The possible values are: 0 : Undefined 2 : Online 3 : Offline 5 : Faulty 6 : Testing
operational-status-string	operational-status-type	The current operational status of the specified port.
enabled-state	uint32	The current state of the specified port. The possible values are: 2 : Enabled. 6 : Disabled. Deprecated: Please use 'is-enabled-state' leaf instead.
is-enabled-state	boolean	The current state of the specified port. The possible values are: true : Enabled. false : Disabled.
user-friendly-name	string	A user-friendly name to be assigned to the specified port. A value from 1 to 128 alphanumeric characters. This name must not contain special (high-ASCII) characters, and is not case-sensitive. When a port name is not configured, a default name is assigned. The format of the default name is as follows: - On standalone platforms, the default port name displays as port<port_number>, for example, port10. - On enterprise-class platforms, the default port name displays as slot<slot_number> port<port_number>, for example, slot1 port5. Provide a null string to replace the existing name with the default.

Name	Type	Description
speed	brocade-interface-types:fc-speed-type	Deprecated: Please reference the protocol-speed leaf instead. The Fibre Channel interface may operate at various speeds; this leaf allows the port interface to be forced to operate at a particular speed. Without any explicit speed set (for example, having the auto-negotiate value set), the Fibre Channel interfaces run at the maximum speed, subject to speed negotiations with their neighbor. The value zero is returned only when auto-negotiate is set and the port has not yet negotiated the speed with the other end (for example, if the port is offline or is syncing). Otherwise, the negotiated speed or the fixed speed is returned. To determine whether the speed was auto-negotiated or fixed, see the auto-negotiate leaf.
protocol-speed	fibrechannel:protocol-speed-type	The Fibre Channel interface may operate at various speeds; this leaf allows the port interface to be forced to operate at a particular speed. Without any explicit speed set (for example, having the auto-negotiate value set), the Fibre Channel interfaces run at the maximum speed, subject to speed negotiations with their neighbor. The value zero is returned only when auto-negotiate is set and the port has not yet negotiated the speed with the other end (for example, if the port is offline or is syncing). Otherwise, the negotiated speed or the fixed speed is returned. To determine whether the speed was auto-negotiated or fixed, see the auto-negotiate leaf. The supported values for protocol-speed are as below.
max-speed	brocade-interface-types:fc-speed-type	Deprecated: Please reference the max-protocol-speed leaf instead. The maximum speed the port is capable of supporting, in bits per second. This is not applicable to Gen7 ports.
max-protocol-speed	fibrechannel:protocol-speed-type	The maximum speed the port is capable of supporting, in bits per second. This is not applicable to Gen7 ports.
auto-negotiate	uint8	Whether the port speed is auto-negotiated on the specified port. 0 : Port speed is fixed. 1 : Port speed is auto-negotiated. This leaf is deprecated. Please use auto-negotiate-v2.
auto-negotiate-v2	boolean	Whether the port speed is auto-negotiated on the specified port. false : Port speed is fixed. true : Port speed is auto-negotiated.
g-port-locked	uint8	Designates the specified port as a G_Port. After successful execution, the switch attempts to initialize the specified port as an F_Port only and does not attempt loop initialization (FL_Port) on the port. A port that is designated as a G_Port can become an E_Port. This configuration can be cleared but not set on VE or VEX_Ports. Changes made by this command are persistent across switch reboots or power cycles. 0 : Unlocked (removes the G_Port designation from the port). 1 : Locked (the port is designated as a G_Port).

Name	Type	Description
e-port-disable	uint8	Enables or disables E_Port capability on the specified port or locks down the specified port as an E_Port. E_Port capability is enabled by default. When an inter-switch link (ISL) is connected to a port and the E_Port capability of the port is disabled, the ISL is segmented, and all traffic between the switches stops. Fabric management data, such as zoning information, can no longer be exchanged through this port. 0 : Enables the port as an E_Port. This is the default port state. 1 : Disables E_Port capability for the specified port. This leaf is deprecated. Please use e-port-disable-v2.
e-port-disable-v2	boolean	Enables or disables E_Port capability on the specified port or locks down the specified port as an E_Port. E_Port capability is enabled by default. When an inter-switch link (ISL) is connected to a port and the E_Port capability of the port is disabled, the ISL is segmented, and all traffic between the switches stops. Fabric management data, such as zoning information, can no longer be exchanged through this port. false : Enables the port as an E_Port. This is the default port state. true : Disables E_Port capability for the specified port.
n-port-enabled	uint8	Indicates whether this port can operate as an N_Port. A port can operate as an N_Port only in Access Gateway Mode. 0 : N_Port is disabled as an N_Port. 1 : N-Port is enabled as an N_Port. This leaf is deprecated. Please use n-port-enabled-v2.
n-port-enabled-v2	boolean	Indicates whether this port can operate as an N_Port. A port can operate as an N_Port only in Access Gateway Mode. false : N_Port is disabled as an N_Port. true : N-Port is enabled as an N_Port.
d-port-enable	uint8	Indicates whether this port is configured as a D_Port. 0 : Port is disabled as a D_Port. 1 : Port is enabled as a D_Port. This leaf is deprecated. Please use d-port-enable-v2.
d-port-enable-v2	boolean	Indicates whether this port is configured as a D_Port. false : Port is disabled as a D_Port. true : Port is enabled as a D_Port.
persistent-disable	uint8	Status of the persistent-disable feature of the port. 0 : Persistent-disable is not active for the port. 1 : Persistent-disable is active for the port. This leaf is deprecated. Please use persistent-disable-v2.
persistent-disable-v2	boolean	Status of the persistent-disable feature of the port. false : Persistent-disable is not active for the port. true : Persistent-disable is active for the port.

Name	Type	Description
application-header-enabled	boolean	Enables or disables the port with application header configuration. On ports where Application Header is not supported, this leaf will not be present in a GET response. Application Header is supported only on Gen7 ports. This leaf is not applicable in FICON enabled switch. This is mutually exclusive with XISL switch configuration. false : Application Header configuration is disabled. true : Application Header configuration is enabled.
locked-e-port-enabled	boolean	Provides the locked E_Port configuration on the switch.. false : locked E_Port configuration is disabled. true : locked E_Port configuration is enabled.
dwdm-losync-enabled	boolean	Provides the dense wavelength-division multiplexing LoSync port configuration on the switch.. false : dense wavelength-division multiplexing LoSync configuration is disabled. true : dense wavelength-division multiplexing LoSync configuration is enabled.
d-port-dwdm-enabled	boolean	Provides the dense wavelength-division multiplexing D_Port configuration on the switch.. false : D_Port dense wavelength-division multiplexing configuration is disabled. true : D_Port dense wavelength-division multiplexing configuration is enabled.
neighbor-slot-port	fibrechannel:slot-port-name-type	The name of the interface. The slot and port number of the port in the format slot/port. A pre-9.1.0 neighbor switch will not have this information.
neighbor-node-wwn	union {fibrechannel:wwn-type;fibrechannel:brocade-empty-string-type;}	The neighbor node WWN connected to this port.
neighbor-port-index	fibrechannel:user-port-number-type	Immediate neighbor port index connected to this port.
neighbor-switch-user-friendly-name	string	The Fibre Channel user-friendly name of the adjacent switch.
fcid	fibrechannel:fcid-type	The Fibre Channel ID (FCID) of the port (decimal format). Deprecated: Please use 'fcid-hex' leaf instead.
fcid-hex	fibrechannel:fcid-hex-string-type	The Fibre Channel ID (FCID) of the port (hexadecimal format).
port-type	brocade-interface-types:port-type-type	The port type currently enabled for the specified port. If this port is logged in, this will be the negotiated port type; otherwise, the configured port type will be reported. Deprecated: Please use 'port-type-string' leaf instead.
port-type-string	brocade-interface-types:port-type-string-type	This attribute represents currently enabled port-type for the specified port. If this port is logged in, this will be the negotiated port type; otherwise, the configured port type will be displayed.

Name	Type	Description
icl-interconnect-status	boolean	Indicates whether the interconnect used is ICL regardless of the port state. true : When the interconnect is ICL. false : When the interconnect is not ICL.
qos-enabled	uint8	Indicates whether QoS is enabled on the port. 0 : Port QoS disabled. 1 : Port QoS enabled. This leaf is deprecated. Please use qos-enabled-v2.
qos-enabled-v2	boolean	Indicates whether QoS is enabled on the port. false : Port QoS disabled. true : Port QoS enabled.
compression-configured	uint8	Enables or disables the compression configuration on the specified port. This change is persistent. Configuring a port for compression is disruptive. You must disable the port before you can enable compression on the port. This command fails on an enabled port. The values are: 0 : Disables the compression configuration on the specified port. 1 : Enables the compression configuration on the specified port. This leaf is deprecated. Please use compression-configured-v2.
compression-configured-v2	boolean	Enables or disables the compression configuration on the specified port. This change is persistent. Configuring a port for compression is disruptive. You must disable the port before you can enable compression on the port. This command fails on an enabled port. The values are: false : Disables the compression configuration on the specified port. true : Enables the compression configuration on the specified port.
encryption-enabled	uint8	Enables or disables the encryption configuration on the port, and saves the change in configuration persistently. Configuring a port for encryption is disruptive. You must disable the port before you can enable compression on the port. This command fails on an enabled port. The values are: 0 - Disables the encryption configuration on the specified port. 1 - Enables the encryption configuration on the specified port. The number of configurable ports is limited by the platform ASIC. This leaf is deprecated. Please use encryption-enabled-v2.
encryption-enabled-v2	boolean	Enables or disables the encryption configuration on the port, and saves the change in configuration persistently. Configuring a port for encryption is disruptive. You must disable the port before you can enable compression on the port. This command fails on an enabled port. The values are: false - Disable the encryption configuration on the specified port. true - Enable the encryption configuration on the specified port. The number of configurable ports is limited by the platform ASIC.
compression-active	uint8	The runtime compression status for the port. 0 : The port is enabled for compression but offline or is not enabled for compression. 1 : The port is online and enabled for compression. This leaf is deprecated. Please use compression-active-v2.

Name	Type	Description
compression-active-v2	boolean	The runtime compression status for the port. false : The port is enabled for compression but offline or is not enabled for compression. true : The port is online and enabled for compression.
encryption-active	uint8	The encryption status for the specified port. 0 : Encryption is disabled. 1 : Encryption is enabled. This leaf is deprecated. Please use encryption-active-v2.
encryption-active-v2	boolean	The encryption status for the specified port. false : Encryption is disabled. true : Encryption is enabled.
target-driven-zoning-enable	uint8	Enables or disables the Target Driven Zoning (TDZ) configuration on the port. This change is persistent. TDZ can be configured on online or offline E Ports, F Ports, and L Ports; it does not toggle the port to apply the configuration. After TDZ is configured on a port, it allows the connected target device to configure Target Driven Peer Zones to be enabled and committed. 0 : Target Driven Zoning configuration is disabled. 1 : Target Driven Zoning configuration is enabled. This leaf is deprecated. Please use target-driven-zoning-enable-v2.
target-driven-zoning-enable-v2	boolean	Enables or disables the Target Driven Zoning (TDZ) configuration on the port. This change is persistent. TDZ can be configured on online or offline E Ports, F Ports, and L Ports; it does not toggle the port to apply the configuration. After TDZ is configured on a port, it allows the connected target device to configure Target Driven Peer Zones to be enabled and committed. false : Target Driven Zoning configuration is disabled. true : Target Driven Zoning configuration is enabled.
sim-port-enabled	uint8	Enables or disables the port as a SIM port. 0 : SIM Port is disabled. 1 : SIM Port is enabled. This leaf is deprecated. Please use sim-port-enabled-v2.
sim-port-enabled-v2	boolean	Enables or disables the port as a SIM port. false : SIM Port is disabled. true : SIM Port is enabled.
mirror-port-enabled	int8	Enables or disables the port as a mirror port. On ports where mirror-port is not supported, this leaf will not be present in a GET response. Port mirroring reroutes data frames between two devices to the mirror port.. 0 : Mirror Port is disabled. 1 : Mirror Port is enabled. This leaf is deprecated. Please use mirror-port-enabled-v2.

Name	Type	Description
mirror-port-enabled-v2	boolean	Enables or disables the port as a mirror port. On ports where mirror-port is not supported, this leaf will not be present in a GET response. Port mirroring reroutes data frames between two devices to the mirror port.. false : Mirror Port is disabled. true : Mirror Port is enabled.
credit-recovery-enabled	int8	Enables or disables credit recovery on the port. Credit recovery enables credits or frames to be recovered. Configure credit recovery on a port. The credit recovery feature enables credits or frames to be recovered. The default credit recovery configuration is enabled. 0 : Credit Recovery is disabled. 1 : Credit Recovery is enabled. This leaf is deprecated. Please use credit-recovery-enabled-v2.
credit-recovery-enabled-v2	boolean	Enables or disables credit recovery on the port. Credit recovery enables credits or frames to be recovered. Configure credit recovery on a port. The credit recovery feature enables credits or frames to be recovered. The default credit recovery configuration is enabled. false : Credit Recovery is disabled. true : Credit Recovery is enabled.
credit-recovery-active	int8	Whether credit recover is active on the port. Fabric OS must support credit recovery at either end of the link for credit recovery to be active. 0 : Credit Recovery is not active on the port. 1 : Credit Recovery is active on the port. This leaf is deprecated. Please use credit-recovery-active-v2.
credit-recovery-active-v2	boolean	Whether credit recover is active on the port. Fabric OS must support credit recovery at either end of the link for credit recovery to be active. false : Credit Recovery is not active on the port. true : Credit Recovery is active on the port.
fec-active	int8	Whether forward error correction (FEC) is active on the port. Fabric OS must support FEC at either end of the link for FEC to be active. 0 : FEC is not active on the port 1 : FEC is active on the port This leaf is deprecated. Please use fec-active-v2.
fec-active-v2	boolean	Whether forward error correction (FEC) is active on the port. Fabric OS must support FEC at either end of the link for FEC to be active. false : FEC is not active on the port true : FEC is active on the port

Name	Type	Description
f-port-buffers	uint8	Configures the buffer allocation for an F_Port. The minimum buffer allocation is the default number of buffers plus 1. The maximum is determined by the remaining buffer allocations in the port's port group. If no buffers are configured, the value is zero. The F_Port buffer feature is not supported on ports configured as EX_Ports, mirror ports, long distance ports, L_Ports, QoS ports, Fast Write, and trunk areas.
e-port-credit	uint8	Configures the number of credits to be allocated to the specified port. The minimum credit allocation is 5, and the maximum can be up to 160 depending on the platform. 0 : E_Port Credit is disabled.
csctl-mode-enabled	int8	Enables or disables Class-Specific Control (CSCTL) mode, which enables traffic prioritization based on CS_CTL. 0 : CSCTL mode is disabled on the port. 1 : CSCTL mode is enabled on the port. This is not applicable for ports on core blades of a chassis. This leaf is deprecated. Please use csctl-mode-enabled-v2.
csctl-mode-enabled-v2	boolean	Enables or disables Class-Specific Control (CSCTL) mode, which enables traffic prioritization based on CS_CTL. false : CSCTL mode is disabled on the port. true : CSCTL mode is enabled on the port. This is not applicable for ports on core blades of a chassis.
fault-delay-enabled	uint8	Configures the fault delay for a port. In the event that the link is noisy after a host power cycle, the switch may go into a soft fault state, which means a delay of R_A_TOV. Setting the mode value to 1 reduces the fault delay value to 1.2 seconds. The configuration is stored in nonvolatile memory and is persistent across switch reboots and power cycles. 0 : The value is R_A_TOV. This is the default. 1 : The value is 1.2 seconds. This leaf is deprecated. Please use fault-delay-enabled-v2.
fault-delay-enabled-v2	boolean	Configures the fault delay for a port. In the event that the link is noisy after a host power cycle, the switch may go into a soft fault state, which means a delay of R_A_TOV. Setting the mode value to 1 reduces the fault delay value to 1.2 seconds. The configuration is stored in nonvolatile memory and is persistent across switch reboots and power cycles. false : The value is R_A_TOV. This is the default. true : The value is 1.2 seconds.

Name	Type	Description
octet-speed-combo	int8	Deprecated: Please use 'octet-speed-combo-string' instead. The speed configuration for a port octet. Valid Octet Combo values include the following: 1 : Auto-negotiated or fixed port speeds of 32Gbps, 16 Gbps, 8 Gbps, 4 Gbps, and 2 Gbps 2 : Auto-negotiated or fixed port speeds of 10 Gbps, 8 Gbps, 4 Gbps, and 2 Gbps 3 : Auto-negotiated or fixed port speeds of 16 Gbps and 10 Gbps A port octet can be set to any of the three octet combinations, and the ports in the octet can run on any speed supported by the port's octet combination. This applies to both auto-negotiated and fixed speeds. The octet combination default is 1.
octet-speed-combo-string	octet-speed-combo-type	The speed configuration for a port octet. Valid Octet Combo values include the following: A port octet can be set to any of the three octet combinations, and the ports in the octet can run on any speed supported by the port's octet combination. This applies to both auto-negotiated and fixed speeds.
long-distance	uint8	Deprecated: Please use 'long-distance-string' instead. The long-distance level. Displays one of the following values (the numerical value representing each distance level is shown in parentheses): 0 : Long distance is disabled for this port. 1 : L0 configures the port as a regular port. 2 : L1 configures the port as long (<= 50 km). 3 : L2 configures the port as super long (<= 100km). 4 : LE configures an E_Port distance as greater than 5 km and up to 10 km. 5 : L0.5 configures the port as medium long (<= 25 km). 6 : LD configures the value as automatic long distance. 7 : LS mode configures the value as a static long distance link with a fixed buffer allocation greater than 10 km.
long-distance-string	long-distance-type	The long-distance level supported for a port.
vc-link-init	uint8	The VC Link initialization. 0 : The long-distance link initialization option is turned off. 1 : The long-distance link initialization option is turned on for long-distance mode. This leaf is deprecated. Please use vc-link-init-v2.
vc-link-init-v2	boolean	The VC Link initialization. false : The long-distance link initialization option is turned off. true : The long-distance link initialization option is turned on for long-distance mode.
isl-ready-mode-enabled	uint8	Enables or disables ISL ready mode on the port. 0 : ISL R_RDY mode is disabled on the port. 1 : ISL R_RDY mode is enabled on the port. This leaf is deprecated. Please use isl-ready-mode-enabled-v2.

Name	Type	Description
isl-ready-mode-enabled-v2	boolean	Enables or disables ISL ready mode on the port. false : ISL R_RDY mode is disabled on the port. true : ISL R_RDY mode is enabled on the port.
rscn-suppression-enabled	uint8	Enables or disables Registered State Change Notification (RSCN) suppression on the port. 0 : Disables RSCN suppression on the port. When disabled, device changes on the port generate an RSCN to all other end devices that are zoned with this one. By default, RSCN suppression is disabled on all ports. 1 : Enables RSCN suppression on the port. When enabled, any device changes on the port do not generate an RSCN to any other end device. This leaf is deprecated. Please use rscn-suppression-enabled-v2.
rscn-suppression-enabled-v2	boolean	Enables or disables Registered State Change Notification (RSCN) suppression on the port. false : Disables RSCN suppression on the port. When disabled, device changes on the port generate an RSCN to all other end devices that are zoned with this one. By default, RSCN suppression is disabled on all ports. true : Enables RSCN suppression on the port. When enabled, any device changes on the port do not generate an RSCN to any other end device.
los-tov-mode-enabled	uint8	Deprecated: Please use 'los-tov-mode-enabled-string' instead. Enables or disables de-bouncing of signal loss for front-end ports for 100 ms. 0 : LOS_TOV mode is disabled on the port. 1 : LOS_TOV mode is enabled on fixed-speed ports. 2 : LOS_TOV mode is enabled on both fixed-speed and auto-negotiated ports.
los-tov-mode-enabled-string	los-tov-mode-type	Enables or disables de-bouncing of signal loss for front-end ports for 100 ms.
npiv-enabled	uint8	Enables or disables NPIV capability on the port. 0 : NPIV is disabled on the port. 1 : NPIV is enabled on the port. This leaf is deprecated. Please use npiv-enabled-v2.
npiv-enabled-v2	boolean	Enables or disables NPIV capability on the port. false : NPIV is disabled on the port. true : NPIV is enabled on the port.
npiv-pp-limit	int16	Configures the maximum number of allowed logins for the port. Valid values are 1 to 255. 1 - 255 : The maximum number of allowed logins for the port.

Name	Type	Description
npiv-flogi-logout-enabled	int8	Whether the base device logout feature is enabled or disabled. 0 : Base device logout is disabled on the port, which causes the NPIV devices on the same port to log out when the base device logs out. 1 : Base device logout is enabled on the port. The base device can log out without disrupting the NPIV devices on the same port. This is not applicable for ports on core blades of a chassis. This leaf is deprecated. Please use npiv-flogi-logout-enabled-v2.
npiv-flogi-logout-enabled-v2	boolean	Whether the base device logout feature is enabled or disabled. false : Base device logout is disabled on the port, which causes the NPIV devices on the same port to log out when the base device logs out. true : Base device logout is enabled on the port. The base device can log out without disrupting the NPIV devices on the same port. This is not applicable for ports on core blades of a chassis.
ex-port-enabled	uint8	Configures the port as an EX_Port. 0 : The port is not configured as an EX_Port. 1 : The port is configured as an EX_Port. This leaf is deprecated. Please use ex-port-enabled-v2.
ex-port-enabled-v2	boolean	Configures the port as an EX_Port. false : The port is not configured as an EX_Port. true : The port is configured as an EX_Port.
fc-router-port-cost	uint32	Specifies the new cost of the link connected to the specified EX_Port. The router module chooses the router port path based on the minimum cost per fabric ID (FID) connection. If multiple paths exist with the same minimum cost, the load is shared over these paths. The new link cost is updated non-disruptively. In case of trunked ports, the link cost update is allowed only on the controller EX_Port trunk ports. The agent trunk ports will adapt the cost value of the controller trunk port.
edge-fabric-id	fibrechannel:fabric-id-type	Specifies the fabric ID of the destination fabric. The valid values for FID are from 1 through 128.
destination-fabric-principal-switch-wwn	fibrechannel:wwn-type	The world wide name of the principal switch in the destination fabric. In case of trunked EX_Ports, this leaf will return empty string for agent ports.
preferred-front-domain-id	fibrechannel:domain-id-type	Specifies the preferred domain ID. The valid values are 1 to 239.
neighbor-switch-ipv4-address	union {inet:ipv4-address;fibrechannel:brocade-empty-string-type;}	The IPv4 address of the switch connected to the EX_Port. In case of trunked EX_Ports, this leaf will return empty string for agent ports.
neighbor-switch-ipv6-address	union {inet:ipv6-address;fibrechannel:brocade-empty-string-type;}	The IPv6 address of the switch connected to the EX-port. In case of trunked EX_Ports, this leaf will return empty string for agent ports.

Name	Type	Description
neighbor-fabric-name	string	Fabric name of the neighboring switch. In case of trunked EX_Ports, this leaf will return empty string for agent ports.
fec-enabled	int8	Whether Forward Error Correction (FEC) is enabled or disabled on a port. FEC provides a mechanism for reducing error rates during data transmissions over 16 Gbps Fibre Channel links. When FEC is enabled on a port, the sender adds systematically generated error-correcting code (ECC) to its data transmission. This mechanism allows the receiver to detect and correct errors without needing to get additional information from the sender. 0 : FEC is disabled on the port. 1 : FEC is enabled on the port, and the port is online. FEC is supported only at 10 Gbps and 16 Gbps speeds. This leaf is deprecated. Please use fec-enabled-v2.
fec-enabled-v2	boolean	Whether Forward Error Correction (FEC) is enabled or disabled on a port. FEC provides a mechanism for reducing error rates during data transmissions over 16 Gbps Fibre Channel links. When FEC is enabled on a port, the sender adds systematically generated error-correcting code (ECC) to its data transmission. This mechanism allows the receiver to detect and correct errors without needing to get additional information from the sender. false : FEC is disabled on the port. true : FEC is enabled on the port, and the port is online. FEC is supported only at 10 Gbps and 16 Gbps speeds.
via-tts-fec-enabled	int8	Whether FEC negotiation via TTS is enabled or disabled on the port. 0 : The external control of FEC is disabled. 1 : The control of the FEC state is permitted via TTS by an externally attached host or device. This leaf is deprecated. Please use via-tts-fec-enabled-v2.
via-tts-fec-enabled-v2	boolean	Whether FEC negotiation via TTS is enabled or disabled on the port. false : The external control of FEC is disabled. true : The control of the FEC state is permitted via TTS by an externally attached host or device.
port-autodisable-enabled	uint8	Enables or disables port autodisable on the port, which minimizes traffic disruption introduced in some instances of automatic port recovery. This parameter disables the port when it encounters an event that would cause the port to reinitialize. When the autodisable flag is enabled, you can specify the conditions that prevent the port to reinitialize. Such conditions include loss of sync, loss of signal, OLS, NOS, and LIP. Note that a link reset does not cause a port autodisable. When a port is in FICON Management Server (FMS) mode, an autodisabled port remains persistently disabled across High Availability (HA) failover. 0 : Port autodisable is disabled on the port. 1 : Port autodisable is enabled on the port. This leaf is deprecated. Please use port-autodisable-enabled-v2.

Name	Type	Description
port-autodisable-enabled-v2	boolean	Enables or disables port autodisable on the port, which minimizes traffic disruption introduced in some instances of automatic port recovery. This parameter disables the port when it encounters an event that would cause the port to reinitialize. When the autodisable flag is enabled, you can specify the conditions that prevent the port to reinitialize. Such conditions include loss of sync, loss of signal, OLS, NOS, and LIP. Note that a link reset does not cause a port autodisable. When a port is in FICON Management Server (FMS) mode, an autodisabled port remains persistently disabled across High Availability (HA) failover. false : Port autodisable is disabled on the port. true : Port autodisable is enabled on the port.
rate-limit-enabled	int16	Configures the rate limits in Mbps. This is applicable only with F/FL_Ports. For E/EX_Ports, this resource is not effective. The ingress rate limit is enforced only when a given port can run at a speed higher than the speed specified in the configuration. Positive numeric values indicate a rate limit in Mbps. 0 : No rate limit set.
non-dfe-enabled	uint8	Enables or disables non-DFE mode on the port. Active receiver DFE uses sophisticated algorithms to automatically adjust the receiver to compensate for signal distortions. 0 : Non-DFE mode is disabled on the port. Automatic receiver adjustment through DFE is activated 1 : Non-DFE mode is enabled on the port. If the non-DFE enabled port is connected to a port that does not require fixed receiver equalization, CRC errors may be detected on the port and the link may toggle. Even if disabled, non-DFE is still automatically activated if both of following conditions are met: the port speed is 8G or N8 and the fillword is IDLE. This is supported only on GEN-5 platforms.
trunk-port-enabled	uint8	Enables or disables trunking on a port. 0 : Trunking is disabled on the port. 1 : Trunking is enabled on the port. Trunking is enabled by default when a trunking license is present on the switch. This leaf is deprecated. Please use trunk-port-enabled-v2.
trunk-port-enabled-v2	boolean	Enables or disables trunking on a port. false : Trunking is disabled on the port. true : Trunking is enabled on the port. Trunking is enabled by default when a trunking license is present on the switch.
default-index	fibrechannel:user-port-number-type	Default Port Index of the port. The default port index can be a port swapped area.
physical-state	string	The physical state of a port.

Name	Type	Description
pod-license-status	boolean	Indicates POD license status for a port. true : The POD license is enabled on the port. false : The POD license is disabled on the port. On Chassis platforms, this leaf returns false. Deprecated: Please use 'pod-license-state' instead.
pod-license-state	enumeration	POD license status of the logical port.
index	fibrechannel:user-port-number-type	The user port number of the front-end port.
reserved-buffers	uint16	Max amount of buffers or reserved buffers if offline. The value increases or decreased when set using the field f-port-buffers and is effective when port is online. If port is offline, the default value is assigned. The max value is derived from the leaf chip-buffers-available. If the value set is too high, it depletes the buffers available to other front end ports on that chip.
average-transmit-buffer-usage	uint16	Average real time buffer usage in Tx path which is calculated for a sampling interval of two seconds.
average-transmit-frame-size	uint16	Average frame size used for transmit including the FC header in bytes which is calculated for a sampling interval of two seconds.
average-receive-buffer-usage	uint16	Average real time buffer usage in Rx path which is calculated for a sampling interval of two seconds.
average-receive-frame-size	uint16	Average frame size used for receive including the FC header in bytes which is calculated for a sampling interval of two seconds.
current-buffer-usage	uint16	Real time usage of buffers by a front end port.
recommended-buffers	uint16	Real time usage of buffers needed by a port to optimally use a link bandwidth. If the value is too less, the port will starve for buffers. If too high, buffers would be wasted and not available to other front end ports on the same chip.
measured-link-distance	string	Actual link distance measured in kilometers.
chip-instance	uint16	The index of the chip on which the given port exists on the chip. Typically used for correlating buffers used by a port to the corresponding chip.
chip-buffers-available	uint16	The front end port on the chip whose available buffer count is being queried.
port-health	string	Information on port health e.g. Offline.
authentication-protocol	string	If authentication is enabled, what authentication protocol is configured on the port.
disable-reason	string	Textual reason why the port is disabled, either defined by the system or by the user. If a user defined reason is set, length is limited to 16 characters. Replay of the same value is idempotent. Only a user defined reasons can be overwritten via REST. One exception, the special system defined value of 'None' can be overwritten by a user defined reason. PATCH operations with an empty string clears a user defined reason.

Name	Type	Description
le-domain	fibrechannel:domain-id-type	Translate domain ID for the port.
port-peer-beacon-enabled	boolean	true : Enabled. false : Disabled.
clean-address-enabled	boolean	Enables or disables Clean Address Bit support for the specified port. The possible values are: true : Enabled. false : Disabled.
congestion-signal-enabled	boolean	Enables or disables Congestion Signal support for the specified port. Only available on Gen7, or later, platforms. The possible values are: true : Enabled. false : Disabled.
segmentation-reason	string	When a port is segmented, this will display the reason for port-segmentation.
lag-name	string	when a fibrechannel interface port is a member of fibrechannel LAG the name of the fibrechannel LAG will be shown.
lag-member-link-status	fc-lag-link-status-type	Indicates operational link status of the fibrechannel LAG member.
user-bound-enabled	boolean	This indicates if the port is bound by user
address-binding-mode	port-address-binding-mode	Indicates if the address binding mode for the port
ms-acl-application-server-access	brocade-interface-types:ms-acl-access-type	The access of Application Server for the specified port.
ms-acl-enhanced-fabric-configuration-server-access	brocade-interface-types:ms-acl-access-type	The access of Enhanced Fabric Configuration Server for the specified port.
ms-acl-fabric-configuration-server-access	brocade-interface-types:ms-acl-access-type	The access of Fabric Configuration Server for the specified port.
ms-acl-fabric-device-management-interface-access	brocade-interface-types:ms-acl-access-type	The access of Fabric Device Management Interface for the specified port.
ms-acl-fabric-zone-server-access	brocade-interface-types:ms-acl-access-type	The access of Fabric Zone Server for the specified port.
ms-acl-unzoned-name-server-access	brocade-interface-types:ms-acl-access-type	The access of Unzoned Name Server for the specified port.
port-generation-number	uint32	The port's generation number for the last offline state change.
port-scn	port-scn-type	The port's last State Change Notification Port-scn can have any of the above enum values.
neighbor	Container	A list of WWN's.
- wwn	fibrechannel:wwn-type	The Fibre Channel WWN of the neighbor port.
areas	Container	A list of 16 bit areas associated with a port on Gen7 platforms or a FCoE port.
- area	fibrechannel:area-address-type	Area of a device.
bound-address-list	Container	The address list that the port is associated with.

Name	Type	Description
- bound-address	fibrechannel:area-address-type	Address bound to the port in hex format.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "fibrechannel": [
            {
                "name": "0/2",
                "wwn": "20:02:d8:1f:cc:5a:f8:4e",
                "port-type": 11,
                "port-type-string": "universal-port",
                "icl-interconnect-status": false,
                "speed": 64000000000,
                "protocol-speed": "64-gfc",
                "user-friendly-name": "port2",
                "operational-status": 3,
                "operational-status-string": "offline",
                "enabled-state": 2,
                "is-enabled-state": true,
                "auto-negotiate": 1,
                "auto-negotiate-v2": true,
                "isl-ready-mode-enabled": 0,
                "isl-ready-mode-enabled-v2": false,
                "long-distance": 0,
                "long-distance-string": "disabled",
                "trunk-port-enabled": 1,
                "trunk-port-enabled-v2": true,
                "vc-link-init": 0,
                "vc-link-init-v2": false,
                "measured-link-distance": "-",
                "pod-license-status": true,
                "pod-license-state": "reserved",
                "index": 2,
                "default-index": 2,
                "fcid": 15598080,
                "fcid-hex": "0xee0200",
```

```
"physical-state": "no_module",
"persistent-disable": 0,
"persistent-disable-v2": false,
"locked-e-port-enabled": false,
"e-port-disable": 0,
"e-port-disable-v2": false,
"ex-port-enabled": 0,
"ex-port-enabled-v2": false,
"dwdm-losync-enabled": false,
"npiv-enabled": 1,
"npiv-enabled-v2": true,
"npiv-pp-limit": 126,
"qos-enabled": 1,
"qos-enabled-v2": true,
"csctl-mode-enabled": 0,
"csctl-mode-enabled-v2": false,
"port-autodisable-enabled": 0,
"port-autodisable-enabled-v2": false,
"e-port-credit": 0,
"d-port-enable": 0,
"d-port-enable-v2": false,
"encryption-enabled": 0,
"encryption-enabled-v2": false,
"d-port-dwdm-enabled": false,
"compression-configured": 0,
"compression-configured-v2": false,
"compression-active": 0,
"compression-active-v2": false,
"encryption-active": 0,
"encryption-active-v2": false,
"mirror-port-enabled": 0,
"mirror-port-enabled-v2": false,
"application-header-enabled": false,
"fec-enabled": 1,
"fec-enabled-v2": true,
"fec-active": 0,
"fec-active-v2": false,
"via-tts-fec-enabled": 0,
"via-tts-fec-enabled-v2": false,
"port-health": "offline",
"authentication-protocol": "none",
"port-peer-beacon-enabled": false,
"neighbor": {},
"areas": {},
"le-domain": 0,
"target-driven-zoning-enable": 0,
"target-driven-zoning-enable-v2": false,
"npiv-flogi-logout-enabled": 0,
"npiv-flogi-logout-enabled-v2": false,
"sim-port-enabled": 0,
"sim-port-enabled-v2": false,
"f-port-buffers": 0,
"clean-address-enabled": false,
```

```
        "congestion-signal-enabled": true,
        "reserved-buffers": 28,
        "average-transmit-buffer-usage": 0,
        "average-transmit-frame-size": 0,
        "average-receive-buffer-usage": 0,
        "average-receive-frame-size": 0,
        "current-buffer-usage": 0,
        "recommended-buffers": 0,
        "chip-instance": 0,
        "chip-buffers-available": 18902,
        "fault-delay-enabled": 0,
        "fault-delay-enabled-v2": false,
        "credit-recovery-enabled": 1,
        "credit-recovery-enabled-v2": true,
        "credit-recovery-active": 0,
        "credit-recovery-active-v2": false,
        "rscn-suppression-enabled": 0,
        "rscn-suppression-enabled-v2": false,
        "los-tov-mode-enabled": 0,
        "los-tov-mode-enabled-string": "disabled",
        "bound-address-list": {
            "bound-address": [
                "0x0200"
            ]
        },
        "user-bound-enabled": false,
        "address-binding-mode": "10-bit",
        "ms-acl-application-server-access": "read-write",
        "ms-acl-enhanced-fabric-configuration-server-access": "read-write",
        "ms-acl-fabric-configuration-server-access": "read-write",
        "ms-acl-fabric-device-management-interface-access": "read-write",
        "ms-acl-fabric-zone-server-access": "read-write",
        "ms-acl-unzoned-name-server-access": "read-only",
        "port-generation-number": 0,
        "port-scn": "offline"
    },
    ...
    {
        "name": "0/63",
        "wwn": "20:3f:d8:1f:cc:5a:f8:4e",
        "port-type": 11,
        "port-type-string": "universal-port",
        "icl-interconnect-status": false,
        "speed": 64000000000,
        "protocol-speed": "64-gfc",
        "user-friendly-name": "port63",
        "operational-status": 3,
        "operational-status-string": "offline",
        "enabled-state": 2,
        "is-enabled-state": true,
        "auto-negotiate": 1,
        "auto-negotiate-v2": true,
        "isl-ready-mode-enabled": 0,
```

```
"isl-ready-mode-enabled-v2": false,
"long-distance": 0,
"long-distance-string": "disabled",
"trunk-port-enabled": 1,
"trunk-port-enabled-v2": true,
"vc-link-init": 0,
"vc-link-init-v2": false,
"measured-link-distance": "-",
"pod-license-status": true,
"pod-license-state": "reserved",
"index": 63,
"default-index": 63,
"fcid": 15613696,
"fcid-hex": "0xee3f00",
"physical-state": "no_module",
"persistent-disable": 0,
"persistent-disable-v2": false,
"locked-e-port-enabled": false,
"e-port-disable": 0,
"e-port-disable-v2": false,
"ex-port-enabled": 0,
"ex-port-enabled-v2": false,
"dwdm-losync-enabled": false,
"npiv-enabled": 1,
"npiv-enabled-v2": true,
"npiv-pp-limit": 126,
"qos-enabled": 1,
"qos-enabled-v2": true,
"csctl-mode-enabled": 0,
"csctl-mode-enabled-v2": false,
"port-autodisable-enabled": 0,
"port-autodisable-enabled-v2": false,
"e-port-credit": 0,
"d-port-enable": 0,
"d-port-enable-v2": false,
"encryption-enabled": 0,
"encryption-enabled-v2": false,
"d-port-dwdm-enabled": false,
"compression-configured": 0,
"compression-configured-v2": false,
"compression-active": 0,
"compression-active-v2": false,
"encryption-active": 0,
"encryption-active-v2": false,
"mirror-port-enabled": 0,
"mirror-port-enabled-v2": false,
"application-header-enabled": false,
"fec-enabled": 1,
"fec-enabled-v2": true,
"fec-active": 0,
"fec-active-v2": false,
"via-tts-fec-enabled": 0,
"via-tts-fec-enabled-v2": false,
```

```
                "port-health": "offline",
                "authentication-protocol": "none",
                "port-peer-beacon-enabled": false,
                "neighbor": {},
                "areas": {},
                "le-domain": 0,
                "target-driven-zoning-enable": 0,
                "target-driven-zoning-enable-v2": false,
                "npiv-flogi-logout-enabled": 0,
                "npiv-flogi-logout-enabled-v2": false,
                "sim-port-enabled": 0,
                "sim-port-enabled-v2": false,
                "f-port-buffers": 0,
                "clean-address-enabled": false,
                "congestion-signal-enabled": true,
                "reserved-buffers": 28,
                "average-transmit-buffer-usage": 0,
                "average-transmit-frame-size": 0,
                "average-receive-buffer-usage": 0,
                "average-receive-frame-size": 0,
                "current-buffer-usage": 0,
                "recommended-buffers": 0,
                "chip-instance": 0,
                "chip-buffers-available": 18902,
                "fault-delay-enabled": 0,
                "fault-delay-enabled-v2": false,
                "credit-recovery-enabled": 1,
                "credit-recovery-enabled-v2": true,
                "credit-recovery-active": 0,
                "credit-recovery-active-v2": false,
                "rscn-suppression-enabled": 0,
                "rscn-suppression-enabled-v2": false,
                "los-tov-mode-enabled": 0,
                "los-tov-mode-enabled-string": "disabled",
                "bound-address-list": {
                    "bound-address": [
                        "0x3f00"
                    ]
                },
                "user-bound-enabled": false,
                "address-binding-mode": "10-bit",
                "ms-acl-application-server-access": "read-write",
                "ms-acl-enhanced-fabric-configuration-server-access": "read-write",
                "ms-acl-fabric-configuration-server-access": "read-write",
                "ms-acl-fabric-device-management-interface-access": "read-write",
                "ms-acl-fabric-zone-server-access": "read-write",
                "ms-acl-unzoned-name-server-access": "read-only",
                "port-generation-number": 0,
                "port-scn": "offline"
            }
        ]
    }
}
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-interface/fibrechannel-lag

Description

The list of fibrechannel LAG interfaces on the switch, their

related configuration and operational state.

Parameters

Table 85: brocade-interface/fibrechannel-lag

Name	Type	Description
name	string	The name of the fibrechannel LAG interface.
number	int32	The port-number of the fibrechannel LAG interface.
port-index	fibrechannel:user-port-number-type	The virtual port index created and assigned to the fibrechannel LAG interface.
port-wwn	fibrechannel:wwn-type	The world wide name (WWN) of the fibrechannel LAG interface.
admin-state-enabled	boolean	Indicates the admin state of the fibrechannel LAG interface. The possible values are: false : fibrechannel LAG is disabled. true : fibrechannel LAG is enabled.
link-status	fc-lag-link-status-type	Operational link status of the fibrechannel LAG interface.
member-ports	Container	The fibrechannel LAG members.
- port	fibrechannel:slot-port-name-type	The list of fibrechannel ports that are members of the fibrechannel LAG interface.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-interface/fibrechannel-router-statistics

Description

Resource usage statistics of active EX_Ports.

Parameters

Table 86: brocade-interface/fibrechannel-router-statistics

Name	Type	Description
port-index	fibrechannel:port-number-type	EX_Port number of the switch.
proxy-device-count	uint32	Displays currently used proxy device for the EX_port.
nr-port-count	uint32	Displays current usage count of NR_Port entries for the physical port(EX_Port).

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-router-statistics>
        <port-index>248</port-index>
        <proxy-device-count>6</proxy-device-count>
        <nr-port-count>2</nr-port-count>
    </fibrechannel-router-statistics>
    <fibrechannel-router-statistics>
        <port-index>249</port-index>
        <proxy-device-count>6</proxy-device-count>
        <nr-port-count>2</nr-port-count>
```

```
    </fibrechannel-router-statistics>
</Response>
```

brocade-interface/fibrechannel-statistics

Description

Statistics for all interfaces on the device.

System-controlled interfaces created by the system are always present in this list, whether they are configured or not.

Parameters

Table 87: brocade-interface/fibrechannel-statistics

Name	Type	Description
name	fibrechannel:slot-port-name-type	The name of the interface. The slot and port number of the port in the format slot/port.
sampling-interval	uint16	The sampling interval in seconds.
time-generated	fibrechannel:time-generated-type	The time at which the statistics were queried.
time-refreshed	fibrechannel:time-generated-type	The time at which the last update of statistics has happened.
reset-statistics	uint8	Deprecated: Please reference brocade-operation-port; port RPC port-reset-statistics instead. Write only leaf. Set to 1 to reset statistical counters.
in-octets	yang:zero-based-counter64	The total number of octets received on the interface, including framing characters.
out-octets	yang:zero-based-counter64	The total number of octets transmitted out of the interface, including framing characters.
in-multicast-pkts	yang:zero-based-counter64	The number of packets, delivered by this sublayer to a higher sublayer, that were addressed to a multicast address at this sublayer. For a MAC-layer protocol, this includes both Group and Functional addresses.
out-multicast-pkts	yang:zero-based-counter64	The total number of packets that higher-level protocols requested be transmitted and that were addressed to a multicast address at this sublayer, including those that were discarded or not sent. For a MAC-layer protocol, this includes both Group and Functional addresses.
in-link-resets	yang:zero-based-counter64	The number of link resets received.
out-link-resets	yang:zero-based-counter64	The total number of link resets transmitted.
in-offline-sequences	yang:zero-based-counter64	The total number of offline sequences received.
out-offline-sequences	yang:zero-based-counter64	The total number of offline sequences transmitted.
invalid-ordered-sets	yang:zero-based-counter64	The total number of invalid ordered sets received.

Name	Type	Description
frames-too-long	yang:zero-based-counter64	The number of frames received that were longer than 2140 octets. The value of 2140 is calculated based on an assumption of 24 header bytes plus 4 CRC bytes and 2112 bytes of payload.
truncated-frames	yang:zero-based-counter64	The total number of truncated frames received.
address-errors	yang:zero-based-counter64	Count of frames received with unknown addressing. An example is an unknown SID or DID, which are not known to the routing algorithm.
delimiter-errors	yang:zero-based-counter64	Count of invalid frame delimiters that are received at this port. An example is a frame with a class 2 at the start and a class 3 at the end.
encoding-disparity-errors	yang:zero-based-counter64	The total number of disparity errors received at this port.
too-many-rdys	yang:zero-based-counter64	Number of times R_RDY was high priority.
in-crc-errors	yang:zero-based-counter64	The number of CRC errors for all frames received.
crc-errors	yang:zero-based-counter64	The number of times that the CRC in a frame does not match the CRC that is computed by the receiver. This count is part of the Link Error Status Block (LESB).
bad-eofs-received	yang:zero-based-counter64	The number of bad EOF frames received.
encoding-errors-outside-frame	yang:zero-based-counter64	The number of encoding-error or disparity-error outside frames received.
multicast-timeouts	yang:zero-based-counter64	The number of multicast frames that have timed out.
in-lcs	yang:zero-based-counter64	The number of link control frames received.
in-frame-rate	yang:zero-based-counter64	The instantaneous receive frame rate.
out-frame-rate	yang:zero-based-counter64	The instantaneous transmit frame rate.
in-max-frame-rate	yang:zero-based-counter64	The maximum frame receive rate since the last reset.
out-max-frame-rate	yang:zero-based-counter64	The maximum frame transmit rate since the last reset.
in-rate	yang:zero-based-counter64	The instantaneous word receive rate.
out-rate	yang:zero-based-counter64	The instantaneous word transmit rate.
in-peak-rate	yang:zero-based-counter64	The peak word receive rate.
out-peak-rate	yang:zero-based-counter64	The peak word transmit rate.
in-frames	yang:zero-based-counter64	The number of frames received at this port.
out-frames	yang:zero-based-counter64	The number of frames transmitted from this port.
bb-credit-zero	yang:zero-based-counter64	The number of transitions in and out of the BB credit zero state.
input-buffer-full	yang:zero-based-counter64	The number of transitions in and out of the Input Buffer Full state.
f-busy-frames	yang:zero-based-counter64	The number of F_BSY (fabric busy) frames generated.
p-busy-frames	yang:zero-based-counter64	The number of P_BSY (port busy) frames generated.
f-rjt-frames	yang:zero-based-counter64	The number of F_RJT (fabric frame reject) frames generated.
p-rjt-frames	yang:zero-based-counter64	The number of P_RJT (port frame reject) frames generated.
class-1-frames	yang:zero-based-counter64	The number of Class 1 frames received at this port.
class-2-frames	yang:zero-based-counter64	The number of Class 2 frames received at this port.
class-3-frames	yang:zero-based-counter64	The number of Class 3 frames received at this port.
class-3-discards	yang:zero-based-counter64	The number of Class 3 frames discarded by the port.

Name	Type	Description
link-failures	yang:zero-based-counter64	The number of link failures at this port.
invalid-transmission-words	yang:zero-based-counter64	The number of invalid transmission words received at this port.
primitive-sequence-protocol-error	yang:zero-based-counter64	The number of primitive sequence protocol errors detected at this port.
loss-of-signal	yang:zero-based-counter64	The number of instances of signal loss detected at this port.
loss-of-sync	yang:zero-based-counter64	The number of instances of synchronization loss detected at this port.
class3-in-discards	yang:zero-based-counter64	The number of class 3 receive frames discarded due to timeout.
class3-out-discards	yang:zero-based-counter64	The number of class 3 transmit frames discarded due to timeout.
pcs-block-errors	yang:zero-based-counter64	The number of Physical Coding Sublayer (PCS) block errors. This counter records encoding violations on 10 Gbps or 16 Gbps ports.
remote-link-failures	yang:zero-based-counter64	The number of link failures at the remote F_Port.
remote-invalid-transmission-words	yang:zero-based-counter64	The number of invalid transmission words received at the remote F_Port.
remote-primitive-sequence-protocol-error	yang:zero-based-counter64	The number of primitive sequence protocol errors detected at the remote F_Port.
remote-loss-of-signal	yang:zero-based-counter64	The number of instances of signal loss detected at the remote F_Port.
remote-loss-of-sync	yang:zero-based-counter64	The number of instances of synchronization loss detected at the remote F_Port.
remote-crc-errors	yang:zero-based-counter64	The number of frames received with an invalid CRC at the remote F_Port.
remote-fec-uncorrected	yang:zero-based-counter64	The counter denotes the number of frames uncorrected by the FEC block at the remote F_Port.
link-level-interrupts	yang:zero-based-counter64	Count of link level interrupts on the port.
frames-processing-required	yang:zero-based-counter64	Count of frames which required processing on the port.
frames-timed-out	yang:zero-based-counter64	Count of frames which timeout out during transmit on the port.
frames-transmitter-unavailable-errors	yang:zero-based-counter64	Count of frames which could not be transmitted due to transmitter not being available.
non-operational-sequences-in	yang:zero-based-counter64	Count of Non Operational Sequences Received.
non-operational-sequences-out	yang:zero-based-counter64	Count of Non Operational Sequences transmitted.
fec-uncorrected	yang:zero-based-counter64	The counter denotes the number of frames uncorrected by the FEC block for the port. The value is zero when fec-active-v2 is false
total-up-time	decimal64	Percent of time the port was up.
total-down-time	decimal64	Percent of time the port was faulty.
down-occurrence	yang:zero-based-counter32	Number of times the port was faulty.
total-offline-time	decimal64	Percent of time the port was offline.
encoding-error-in	yang:zero-based-counter64	The counter denotes the number of encoding errors inside of frames for the port.
tx-peak-frame	yang:zero-based-counter64	The counter denotes the tx peak frame rate in frames/second for the port.

Name	Type	Description
rx-peak-frame	yang:zero-based-counter64	The counter denotes the rx peak frame rate in frames/second for the port.
starvation-tenancy-stop	yang:zero-based-counter64	The number of loop tenancies stopped due to starvation.
invalid-arbitrate-loop	yang:zero-based-counter64	The number of invalid arbitrated loops for the port.
ftb-type-1-miss	yang:zero-based-counter64	The number of FCR frames with transmit errors on the port.
ftb-type-2-miss	yang:zero-based-counter64	The number of FCR frames with routing errors on the port.
ftb-type-6-miss	yang:zero-based-counter64	The number of FCR frames with receive errors on the port.
hard-zone-miss	yang:zero-based-counter64	The number of frames with hard zoning miss on the port.
lun-zone-miss	yang:zero-based-counter64	The number of frames with logical unit number (LUN) zoning miss on the port.
state-transition-count	uint32	The total number of state transitions on a specified port.
interrupt-count	uint32	The total number of interrupts on a port.
unknown-interrupt-count	uint32	Total number of unknown interrupts that are not counted elsewhere.
congestion-primitive-in	uint32	Number of congestion detection signals received by the switch from the device.
remote-buffer-credit-info	Container	The buffer credit values. Applicable only for the remote F_Port.
- bb-credit	yang:zero-based-counter64	The buffer-to-buffer (BB) credit represents the number of credits available to the attached devices.
- peer-bb-credit	yang:zero-based-counter64	The peer port buffer-to-buffer(BB) credit represents the number of credits available to the switch port.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-statistics>
        <name>7/0</name>
        <in-link-resets>2</in-link-resets>
        <out-link-resets>4</out-link-resets>
        <out-offline-sequences>2</out-offline-sequences>
        <in-offline-sequences>0</in-offline-sequences>
        <invalid-ordered-sets>0</invalid-ordered-sets>
        <frames-too-long>0</frames-too-long>
        <truncated-frames>0</truncated-frames>
```

```
<address-errors>0</address-errors>
<delimiter-errors>0</delimiter-errors>
<encoding-disparity-errors>0</encoding-disparity-errors>
<too-many-rdys>0</too-many-rdys>
<in-crc-errors>0</in-crc-errors>
<bad-eofs-received>0</bad-eofs-received>
<encoding-errors-outside-frame>0</encoding-errors-outside-frame>
<multicast-timeouts>0</multicast-timeouts>
<in-lcs>14948</in-lcs>
<out-frame-rate>0</out-frame-rate>
<in-frame-rate>0</in-frame-rate>
<out-max-frame-rate>200</out-max-frame-rate>
<in-max-frame-rate>1</in-max-frame-rate>
<out-rate>0</out-rate>
<in-rate>0</in-rate>
<out-peak-rate>5649</out-peak-rate>
<in-peak-rate>15</in-peak-rate>
<out-frames>177639</out-frames>
<in-frames>29901</in-frames>
<out-octets>27988096</out-octets>
<in-octets>1696876</in-octets>
<bb-credit-zero>124856398</bb-credit-zero>
<input-buffer-full>0</input-buffer-full>
<f-busy-frames>0</f-busy-frames>
<p-busy-frames>0</p-busy-frames>
<f-rjt-frames>0</f-rjt-frames>
<p-rjt-frames>0</p-rjt-frames>
<class-1-frames>0</class-1-frames>
<class-2-frames>0</class-2-frames>
<class-3-frames>0</class-3-frames>
<class-3-discards>0</class-3-discards>
<in-multicast-pkts>0</in-multicast-pkts>
<out-multicast-pkts>0</out-multicast-pkts>
<link-failures>0</link-failures>
<crc-errors>0</crc-errors>
<invalid-transmission-words>0</invalid-transmission-words>
<primitive-sequence-protocol-error>0</primitive-sequence-protocol-error>
<loss-of-signal>0</loss-of-signal>
<loss-of-sync>0</loss-of-sync>
<link-level-interrupts>22</link-level-interrupts>
<frames-processing-required>29887</frames-processing-required>
<frames-timed-out>0</frames-timed-out>
<frames-transmitter-unavailable-errors>0</frames-transmitter-unavailable-errors>
<sampling-interval>2</sampling-interval>
<time-generated>1636405725</time-generated>
<time-refreshed>1636405725</time-refreshed>
<reset-statistics>0</reset-statistics>
<class3-out-discards>0</class3-out-discards>
<class3-in-discards>0</class3-in-discards>
<fec-uncorrected>0</fec-uncorrected>
<pcs-block-errors>0</pcs-block-errors>
<non-operational-sequences-in>0</non-operational-sequences-in>
<non-operational-sequences-out>0</non-operational-sequences-out>
```

```
        <total-up-time>99.85</total-up-time>
        <total-down-time>0.00</total-down-time>
        <down-occurrence>0</down-occurrence>
        <total-offline-time>0.15</total-offline-time>
    </fibrechannel-statistics>
...
    <fibrechannel-statistics>
        <name>6/31</name>
        <in-link-resets>0</in-link-resets>
        <out-link-resets>0</out-link-resets>
        <out-offline-sequences>0</out-offline-sequences>
        <in-offline-sequences>0</in-offline-sequences>
        <invalid-ordered-sets>0</invalid-ordered-sets>
        <frames-too-long>0</frames-too-long>
        <truncated-frames>0</truncated-frames>
        <address-errors>0</address-errors>
        <delimiter-errors>0</delimiter-errors>
        <encoding-disparity-errors>0</encoding-disparity-errors>
        <too-many-rdys>0</too-many-rdys>
        <in-crc-errors>0</in-crc-errors>
        <bad-eofs-received>0</bad-eofs-received>
        <encoding-errors-outside-frame>0</encoding-errors-outside-frame>
        <multicast-timeouts>0</multicast-timeouts>
        <in-lcs>0</in-lcs>
        <out-frame-rate>0</out-frame-rate>
        <in-frame-rate>0</in-frame-rate>
        <out-max-frame-rate>0</out-max-frame-rate>
        <in-max-frame-rate>0</in-max-frame-rate>
        <out-rate>0</out-rate>
        <in-rate>0</in-rate>
        <out-peak-rate>0</out-peak-rate>
        <in-peak-rate>0</in-peak-rate>
        <out-frames>0</out-frames>
        <in-frames>0</in-frames>
        <out-octets>0</out-octets>
        <in-octets>0</in-octets>
        <bb-credit-zero>0</bb-credit-zero>
        <input-buffer-full>0</input-buffer-full>
        <f-busy-frames>0</f-busy-frames>
        <p-busy-frames>0</p-busy-frames>
        <f-rjt-frames>0</f-rjt-frames>
        <p-rjt-frames>0</p-rjt-frames>
        <class-1-frames>0</class-1-frames>
        <class-2-frames>0</class-2-frames>
        <class-3-frames>0</class-3-frames>
        <class-3-discards>0</class-3-discards>
        <in-multicast-pkts>0</in-multicast-pkts>
        <out-multicast-pkts>0</out-multicast-pkts>
        <link-failures>0</link-failures>
        <crc-errors>0</crc-errors>
        <invalid-transmission-words>0</invalid-transmission-words>
        <primitive-sequence-protocol-error>0</primitive-sequence-protocol-error>
        <loss-of-signal>0</loss-of-signal>
```

```
        <loss-of-sync>0</loss-of-sync>
        <link-level-interrupts>0</link-level-interrupts>
        <frames-processing-required>0</frames-processing-required>
        <frames-timed-out>0</frames-timed-out>
        <frames-transmitter-unavailable-errors>0</frames-transmitter-unavailable-errors>
        <sampling-interval>2</sampling-interval>
        <time-generated>1636405725</time-generated>
        <time-refreshed>1636405725</time-refreshed>
        <reset-statistics>0</reset-statistics>
        <class3-out-discards>0</class3-out-discards>
        <class3-in-discards>0</class3-in-discards>
        <fec-uncorrected>0</fec-uncorrected>
        <pcs-block-errors>0</pcs-block-errors>
        <non-operational-sequences-in>0</non-operational-sequences-in>
        <non-operational-sequences-out>0</non-operational-sequences-out>
        <total-up-time>0.00</total-up-time>
        <total-down-time>0.00</total-down-time>
        <down-occurrence>0</down-occurrence>
        <total-offline-time>100.00</total-offline-time>
    </fibrechannel-statistics>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-interface/gigabitethernet

Description

The list of gigabitethernet interfaces on the device. System-controlled

interfaces created by the system are always present in this list,

whether they are configured or not.

Parameters

Table 88: brocade-interface/gigabitethernet

Name	Type	Description
name	brocade-interface-types:ge-interface-type	The name of the interface.
enabled-state	uint8	Enabled or disabled state of the brocade-interface-types: Please use '../enabled-state-v2' leaf as this leaf is deprecated. 0 : Disabled 1 : Enabled
enabled-state-v2	boolean	Enabled or disabled state of the GE port: false : Disabled true : Enabled

Name	Type	Description
speed	brocade-interface-types:speed-type	Deprecated: Please reference the protocol-speed leaf instead. For PHY types that may operate at various speeds, this leaf allows the interface to be forced to operate at a particular speed. Without any explicit configuration, gigabitethernet interfaces run at the maximum capable speed.
protocol-speed	fibrechannel:protocol-speed-type	For PHY types that may operate at various speeds, this leaf allows the interface to be forced to operate at a particular speed. Without any explicit configuration, gigabitethernet interfaces run at the maximum capable speed. The supported values for protocol-speed are as below. 1-ge 10-ge 25-ge 40-ge 100-ge
fec	brocade-interface-types:fec-type	The GE port Forward Error Correction settings used. supported values: off : Forward Error Correction is disabled for the port cl108: Forward Error Correction clause 108 is set. This is valid for ports with protocol-speed set to 25-ge port speed. cl91 :Forward Error Correction clause 91 is set. This is valid for ports with protocol-speed set to 100-ge port speed
mac-address	yang:mac-address	MAC Address for the interface.
operational-status	uint16	Operational status of the port: 1 : Online 2 : Offline
persistent-disable	uint8	Indicates if the port is persistently disabled or persistently enabled. Please refer leaf '../persistent-disable-v2' as this leaf is deprecated. 1 : persistently disabled 0 : persistently enabled
persistent-disable-v2	boolean	Indicates if the port is persistently disabled or persistently enabled true : persistently disabled false : persistently enabled
protocol	extension:ge-protocol-type	The GE port protocol configuration used by Extension blade/Switch during deployment. When GE port protocol is configured for LAN, the GE port is connected on the LAN side for Extending LAN IP traffic over Extension tunnels using IP QOS priorities. When the GE port protocol is configured for WAN then the GE ports are connected on the WAN side and are used for configuring the FCIP/Extension tunnels using them. In case of chassis system if the slot is powered off or in faulty state then the leaf will be skipped in display. Supported values: LAN : GE port is configured for LAN side connections. WAN : GE port is configured for WAN side connections.

Name	Type	Description
auto-negotiation-enabled	boolean	Auto-negotiation is enabled by default in 1G mode. In 10G mode it is disabled and not supported. When the port is set for 1G mode, you can disable auto-negotiation. In case of chassis system if the slot is powered off or in faulty state then the leaf will be skipped in display. Supported values: true : Auto negotiation mode for GE port is enabled. false : Auto negotiation mode for GE port is disabled.
portchannel-name	brocade-interface-types:portchannel-interface-name-type	The name of the portchannel that the gigabit-ethernet interface belongs to. This will be NULL in case the port is not portchannel member
portchannel-member-timeout	lag-port-timeout	The dynamic portchannel member timeout of the gigabit-ethernet interface. default long.
lldp-profile	string	LLDP profile name configured on the port. Blank lldp-profile means LLDP profile is not configured on this port. In such case, lldp global parameters are in use for this port. To configure a new profile or change existing profile on the port, user should perform a PATCH operation with the profile name. To remove the profile from the port, user should perform a PATCH operation with NULL string.
lldp-enabled-state	boolean	LLDP state of the port. Default state is true. Possible values are: true - LLDP is enabled on the port false - LLDP is disabled on the port
extension-vlans	Container	List of extension vlans configured on this port
- vlan	uint32	Extension vlans configured on this port

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
{
    "Response": {
        "gigabitethernet": [
            {
                "name": "0/0",
                "enabled-state": 1,
```

```
                "enabled-state-v2": true,
                "speed": 10000000000,
                "protocol-speed": "10-ge",
                "mac-address": "c4:f5:7c:02:8b:09",
                "operational-status": 2,
                "persistent-disable": 0,
                "persistent-disable-v2": false,
                "portchannel-name": "",
                "lldp-profile": "",
                "lldp-enabled-state": true,
                "extension-vlans": {},
                "protocol": "WAN",
                "fec": "off"
            },
 ...
            {
                "name": "0/17",
                "enabled-state": 1,
                "enabled-state-v2": true,
                "speed": 100000000000,
                "protocol-speed": "100-ge",
                "mac-address": "c4:f5:7c:02:8b:1a",
                "operational-status": 2,
                "persistent-disable": 0,
                "persistent-disable-v2": false,
                "portchannel-name": "",
                "lldp-profile": "",
                "lldp-enabled-state": true,
                "extension-vlans": {},
                "protocol": "WAN",
                "fec": "cl91"
            }
        ]
    }
}
```


PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-interface/gigabitethernet-statistics

Description

A list of interface-related statistics for gigabitethernet port.

Parameters

Table 89: brocade-interface/gigabitethernet-statistics

Name	Type	Description
name	brocade-interface-types:ge-interface-type	The name of the interface.
out-pkts	yang:zero-based-counter64	The total number of packets transmitted out of the interface.
out-octets	yang:zero-based-counter64	The total number of octets transmitted out of the interface.
out-unicast-pkts	yang:zero-based-counter64	The total number of unicast packets transmitted out of the interface.
out-multicast-pkts	yang:zero-based-counter64	The total number of multicast packets transmitted out of the interface.
out-broadcast-pkts	yang:zero-based-counter64	The total number of broadcast packets transmitted out of the interface.
out-vlan-pkts	yang:zero-based-counter64	The total number of vlan packets transmitted out of the interface.
out-pause-pkts	yang:zero-based-counter64	The total number of pause packets transmitted out of the interface.
in-pkts	yang:zero-based-counter64	The number of packets received on the interface.
in-octets	yang:zero-based-counter64	The total number of octets received on the interface.
in-unicast-pkts	yang:zero-based-counter64	The number of unicast packets received on the interface.
in-multicast-pkts	yang:zero-based-counter64	The number of multicast packets received on the interface.
in-broadcast-pkts	yang:zero-based-counter64	The number of broadcast packets received on the interface.
in-vlan-pkts	yang:zero-based-counter64	The total number of vlan packets received on the interface.
in-pause-pkts	yang:zero-based-counter64	The total number of pause packets received on the interface.
carrier-loss-error	yang:zero-based-counter64	The total number of carrier loss errors.
crc-error	yang:zero-based-counter64	The total number of Cyclic Redundancy Check (CRC) errors.
jabber-error	yang:zero-based-counter64	The total number of jabber errors.
reset-statistics	uint8	Write-only leaf and will not be returned in GET request. Set it to 1 to reset statistical counters in PATCH request. Please use rpc 'brocade-operation-port' to reset the statistics for GE ports.
time-generated	fibrechannel:time-generated-type	The time at which the statistics were queried.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <gigabitethernet-statistics>
  <name>4/0</name>
  <out-pkts>178269</out-pkts>
  <out-octets>23396402</out-octets>
  <out-unicast-pkts>176862</out-unicast-pkts>
  <out-multicast-pkts>0</out-multicast-pkts>
  <out-broadcast-pkts>1407</out-broadcast-pkts>
  <out-vlan-pkts>0</out-vlan-pkts>
  <out-pause-pkts>0</out-pause-pkts>
  <in-pkts>278070</in-pkts>
  <in-octets>30069936</in-octets>
  <in-unicast-pkts>267797</in-unicast-pkts>
  <in-multicast-pkts>0</in-multicast-pkts>
  <in-broadcast-pkts>10273</in-broadcast-pkts>
  <in-vlan-pkts>0</in-vlan-pkts>
  <in-pause-pkts>0</in-pause-pkts>
  <carrier-loss-error>3</carrier-loss-error>
  <crc-error>0</crc-error>
  <jabber-error>0</jabber-error>
  <reset-statistics>0</reset-statistics>
  <time-generated>1500431065</time-generated>
 </gigabitethernet-statistics>
...
 <gigabitethernet-statistics>
  <name>4/17</name>
  <out-pkts>0</out-pkts>
  <out-octets>0</out-octets>
  <out-unicast-pkts>0</out-unicast-pkts>
  <out-multicast-pkts>0</out-multicast-pkts>
  <out-broadcast-pkts>0</out-broadcast-pkts>
  <out-vlan-pkts>0</out-vlan-pkts>
  <out-pause-pkts>0</out-pause-pkts>
  <in-pkts>0</in-pkts>
  <in-octets>0</in-octets>
  <in-unicast-pkts>0</in-unicast-pkts>
  <in-multicast-pkts>0</in-multicast-pkts>
  <in-broadcast-pkts>0</in-broadcast-pkts>
  <in-vlan-pkts>0</in-vlan-pkts>
  <in-pause-pkts>0</in-pause-pkts>
  <carrier-loss-error>0</carrier-loss-error>
  <crc-error>0</crc-error>
  <jabber-error>0</jabber-error>
  <reset-statistics>0</reset-statistics>
  <time-generated>1500431065</time-generated>
 </gigabitethernet-statistics>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-interface/logical-e-port

Description

The list of logical E_Port interfaces on the device which form

the logical inter switch link (lisl).

Parameters

Table 90: brocade-interface/logical-e-port

Name	Type	Description
port-index	fibrechannel:user-port-number-type	The unique port number on the switch for identifying a logical E_Port.
fabric-id	fibrechannel:fabric-id-type	The virtual fabric identification (VFID) of the logical switch in which the logical E_Port is created.
operational-status	enumeration	Operational status of the the logical E_Port.
offline-reason	enumeration	Reason for logical E_Port Offline status.
neighbor-node-wwn	fibrechannel:wwn-type	The neighbor node WWN logically connected to this port.
associated-physical-ports	Container	Associated physical E_Port list.
- port	fibrechannel:slot-port-name-type	Physical port number of base switch associated with logical E_Port.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.


```
  {
 "Response": {
  "logical-e-port": [
   {
    "name": 64,
    "fabric-id": 128,
    "operational-status": 2,
```

```
     "associated-physical-ports": {
     "port": [
     "42",
     "1"
    ]
   }
  }
 ]
 }
 }
```


brocade-interface/portchannel

Description

The list of portchannel interfaces on the switch, their related configuration and operational state.

Parameters

Table 91: brocade-interface/portchannel

Name	Type	Description
name	brocade-interface-types:portchannel-interface-name-type	The portchannel interface name.
key	uint32	The portchannel key. The portchannel key cannot be modified once the portchannel is created. If user does not provide the key during portchannel creation, the system will auto-assign an available key. The portchannel key is unique across all portchannel interfaces on the switch.
portchannel-type	portchannel-type	The type of a portchannel interface. Portchannel Type is a mandatory parameter while the portchannel is created. Possible values: static : Static portchannel type. dynamic : Dynamic portchannel type.
speed	brocade-interface-types:portchannel-speed-type	Deprecated: Please reference the protocol-speed leaf instead. The speed of the portchannel interface. The speed of the portchannel is the speed of its member ports. speed is an read-only parameter of a port-channel. Note: The speed of all members of a portchannel must match.
protocol-speed	fibrechannel:protocol-speed-type	The speed of the portchannel interface. The speed of the portchannel is the speed of its member ports. speed is an read-only parameter of a port-channel. Note: The speed of all members of a portchannel must match.
lacp-system-mac-address	ietf:mac-address	The LACP MAC address of the system.

Name	Type	Description
admin-state-enabled	boolean	Indicates the admin state of the portchannel. The possible values are: false : portchannel is disabled. true : portchannel is enabled.
auto-negotiation-enabled	boolean	Indicates the auto-negotiation of the FCIP portchannel. The possible values are: false : auto-negotiation is off. true : auto-negotiation is on.
operational-status-enabled	boolean	The portchannel operational status. The possible values are: false : portchannel is offline. true : portchannel is online.
primary-member-gigabit-ethernet-port	brocade-interface-types:ge-interface-type	The primary member interface of the portchannel.
gigabit-ethernet-member-ports	Container	The Gigabit Ethernet portchannel members.
- port	brocade-interface-types:ge-interface-type	The list of gigabit ethernet interfaces that are members of the portchannel.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message and a “200 OK” status in the headers.

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-interface/portchannel-statistics

Description

A list of interface-related statistics for portchnannel interface .

Parameters

Table 92: brocade-interface/portchannel-statistics

Name	Type	Description
name	brocade-interface-types:portchannel-interface-name-type	The portchannel interface name.
out-pkts	yang:zero-based-counter64	The total number of packets transmitted out of the interface.
out-octets	yang:zero-based-counter64	The total number of octets transmitted out of the interface.
out-unicast-pkts	yang:zero-based-counter64	The total number of unicast packets transmitted out of the interface.
out-multicast-pkts	yang:zero-based-counter64	The total number of multicast packets transmitted out of the interface.
out-broadcast-pkts	yang:zero-based-counter64	The total number of broadcast packets transmitted out of the interface.
out-vlan-pkts	yang:zero-based-counter64	The total number of vlan packets transmitted out of the interface.
out-pause-pkts	yang:zero-based-counter64	The total number of pause packets transmitted out of the interface.
in-pkts	yang:zero-based-counter64	The number of packets received on the interface.
in-octets	yang:zero-based-counter64	The total number of octets received on the interface.
in-unicast-pkts	yang:zero-based-counter64	The number of unicast packets received on the interface.
in-multicast-pkts	yang:zero-based-counter64	The number of multicast packets received on the interface.
in-broadcast-pkts	yang:zero-based-counter64	The number of broadcast packets received on the interface.
in-vlan-pkts	yang:zero-based-counter64	The total number of vlan packets received on the interface.
in-pause-pkts	yang:zero-based-counter64	The total number of pause packets received on the interface.
carrier-loss-error	yang:zero-based-counter64	The total number of carrier loss errors.
crc-error	yang:zero-based-counter64	The total number of Cyclic Redundancy Check (CRC) errors.
jabber-error	yang:zero-based-counter64	The total number of jabber errors.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-license/end-user-license-agreement

Description

Detailed view of End User License Agreement.

Parameters

Table 93: brocade-license/end-user-license-agreement

Name	Type	Description
text	string	Contents of accepted Brocade EULA (End User License Agreement).

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.


```
{
 "Response": {
  "end-user-license-agreement": {
   "text": "*************************************************************************
 \n* BROCADE FABRIC OPERATING SYSTEM AND FEATURE LICENSES AND LICENSE KEYS *
\n*************************************************************************\n\n\nEND USER LICENSE AGREEMENT
\n==========================\n\nTHIS END USER LICENSE AGREEMENT (\"Agreement\") GOVERNS THE DOWNLOAD,
 INSTALLATION,\nUSE, POSTING, DISTRIBUTING AND OTHERWISE MAKING AVAILABLE OF BROCADE'S FABRIC \nOPERATING
 SYSTEM (\"FOS\") SOFTWARE AND/ OR USE OF BROCADE FEATURE LICENSES AND \nLICENSE KEYS THAT ACTIVATE FOS OR
 FUNCTIONALITY WITHIN FOS, AND ACCOMPANYING \nDOCUMENTATION (collectively the \"Software\"). BY DOWNLOADING,
 INSTALLING, USING, \nPOSTING, DISTRIBUTING OR OTHERWISE MAKING AVAILABLE THE SOFTWARE, OR BY \nPURCHASING,
 CONVERTING A TRANSACTION KEY INTO A LICENSE KEY, OR INSTALLING A \nLICENSE OR LICENSE KEY, YOU ARE AGREEING
 TO BE BOUND ON AN ONGOING BASIS BY THE \nTERMS AND CONDITIONS HEREIN, WHICH MAY BE UPDATED BY BROCADE FROM
 TIME TO TIME. \nIF AT ANY TIME YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, PROMPTLY \nSTOP
 USE OF THE SOFTWARE AND DESTROY ALL COPIES OF THE SOFTWARE IN YOUR \nPOSSESSION OR CONTROL, AND CERTIFY
 IN WRITING TO BROCADE SUCH CESSATION OF USE \nAND DESTRUCTION.\n\nSingle User License\n==================
\nSubject to the terms and conditions of this Agreement and payment of the applicable\nlicense fees, Brocade
 Communications Systems LLC (\"Brocade\") and its suppliers grant\nto you (\"End User\") a non-exclusive,
 non-transferable, non-assignable, non-sub licensable\nlicense to use the Software in object code form
 (in the case of FOS) solely for the\npurpose of operating Brocade storage area networking switches in
 accordance with the\nsection entitled \"Restrictions\" below. End User may make such backup copies of the
\nSoftware as may be necessary for End User's lawful use, provided End User affixes to\nsuch copies all
 trademark, copyright, patent, and notices of other proprietary rights\nthat appear on the original.\n\nFOS
```

Licenses and License Keys\n============================\nFOS Licenses are used to activate FOS features or functionality in Brocade switches.\nFOS Licenses shall only be purchased from a Brocade authorized distributor or official\nOEM partner, and FOS License Keys shall only be obtained through Brocade's License Portal\nor an authorized OEM License Portal and shall only be used on Brocade switches purchased\nthrough authorized channels. FOS License Keys obtained from any other sources are not\npermitted to be used on any Brocade hardware products. FOS Licenses and License Keys\nshall not be resold, assigned, sub-licensed or otherwise transferred to a third party.\n\nOpen Source Software\n====================\nCertain components of the Software may incorporate \"open source\" software. Open source\nsoftware is subject to the applicable open source license (e.g., the GNU General Public\nLicense) and is not subject to this Agreement. To obtain a copy of the source code and\napplicable licensing terms for the open source software used by Brocade, please download\nthe open source disclosure documents in the Broadcom Customer Support Portal (CSP). If\nyou do not have a CSP account or are unable to login, please contact your support\nprovider for this information. \n\nRestrictions\n============\nFOS is designed for use with Brocade hardware products sold through authorized sales\nchannels, and with hardware platforms that are currently supported by Brocade. End User\nacknowledges that use of the Software on hardware products that do not meet the above\ncriteria, or which is inconsistent with the restrictions below, presents risks to the\nsecurity and quality of Products and to the Software, for which Brocade disclaims all\nliability. Except as otherwise expressly provided in this Agreement or expressly\nauthorized by Brocade, End User shall have no right, and End User specifically agrees\nnot to, and not to permit third parties to: (i) modify, adapt, or create derivative\nworks based upon the Software; (ii) copy, in whole or in part, decompile, translate,\nreverse engineer, disassemble or otherwise reduce the Software to humanreadable form;\n(iii) install the Software on any products other than Brocade storage area networking\nswitches purchased by End User through authorized channels for its own use or for \nresale expressly authorized by Brocade; (iv) install or use the Software on products\nin excess of the maximum number of Brocade storage area networking switches for which\nEnd User is licensed; (v) install the Software on any Brocade storage area networking\nswitches that have reached End of Support as defined by Brocade, or refurbished devices,\nexcept with Brocade's prior written consent, (vi) use the Software on any devices not\ndesignated by Brocade for use with the Software (as notified through software release\nnotes and other notices), (vii) download, or make available for use or download, any\nversions of the Software that have reached End of Support, as declared by Brocade, or\n(viii) sub-license or permit third parties to post or otherwise make available, or\nuse the Software to provide support for Brocade products unless expressly authorized\nby Brocade.\n\nUpgrades and Additional Copies\n==============================\nFor purposes of this Agreement, \"Software\" shall also include any upgrades, updates,\nbug fixes or modified versions (\"Upgrades\") provided to End User by Brocade or an\nauthorized distributor and for which End User had paid the applicable license fees,\nand any backup copies of the Software. Notwithstanding the foregoing, End User\nacknowledges and agrees that Brocade and its resellers and distributors shall have\nno obligation to provide any Upgrades. If Upgrades or additional copies of the\nSoftware are provided, End User acknowledges and agrees that it has no license or\nright to use such additional copies or Upgrades unless End User, at the time of\nacquiring such copy or Upgrade, already holds a valid license to the original\nSoftware for the applicable number of copies.\n\nNo Support\n==========\nBrocade may, but is not required to, provide technical support for the Software.\n\nSecurity Notice\n===============\nSoftware may include security features that End User can use, along with other security\ntools, to implement increased security in End User's storage area network (\"SAN\").\nHowever, use of the security features of the Software does not guarantee the security\nof End User's SAN or overall network. There are numerous factors that affect the security\nof a SAN, including, without limitation, correct installation and setup of the security\nfeatures of Software and all related requirements, correctly configured security\npolicies, selection of hardware and software (including network security tools),\ncorrect installation, configuration, and maintenance of the hardware and software,\nthe interoperability of the various components of End User's SAN and End User's network,\nand a physically and electronically secure operating environment. In addition, Software\nmay utilize digital certificates in connection with its access control features.\nAlthough digital certificates are a useful authentication security measure that improves\noverall security, they do not guarantee authenticity or security. In designing the\nsecurity of End User's SAN, it is End User's responsibility to evaluate all of these\nfactors to ensure End User's SAN meets End User's security requirements. End User's\nexperience may vary based on these and other factors.

End User's use of Software,\nincluding the digital certificates, is subject to and governed by the terms of this\nAgreement and to End User's compliance with the policies and procedures for the use\nof the security features of Software and digital certificates, which may be made\navailable to End User by Brocade from time to time. If Brocade becomes aware of a breach\nof the security of its digital certificate infrastructure, Brocade reserves the right\nto re-issue digital certificates. In that event, End User will be required to submit\nnew certificate signing requests and install reissued certificates across End User's\nSAN. End User should plan for any network disruption that this may cause. END USER\nACKNOWLEDGES THAT END USER HAS ACCESS TO SUFFICIENT INFORMATION TO ENSURE THAT END USER\nCAN MAKE AN INFORMED DECISION AS TO THE EXTENT TO WHICH END USER CHOOSES TO RELY ON\nDIGITAL CERTIFICATES AND OTHER SECURITY FEATURES IN SOFTWARE (\"SECURITY\").\n\nDisclaimer of Warranty\n======================\nTO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, BROCADE PROVIDES THE SOFTWARE\n\"AS IS\" WITHOUT WARRANTY OF ANY KIND. BROCADE DOES NOT WARRANT THAT THE SOFTWARE\nIS ERROR FREE OR THAT END USER WILL BE ABLE TO OPERATE THE SOFTWARE WITHOUT\nPROBLEMS OR INTERRUPTIONS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW,\nBROCADE AND ITS LICENSORS, SUPPLIERS, DISTRIBUTORS AND RESELLERS DISCLAIM ALL\nEXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT\nLIMITATION, ANY IMPLIED WARRANTY OR CONDITION OF MERCHANTABILITY, FITNESS FOR A\nPARTICULAR PURPOSE, NONINFRINGEMENT, AND SATISFACTORY QUALITY OR ARISING FROM A\nCOURSE OF DEALING, USAGE, OR TRADE PRACTICE.\n\nLimitation of Liability\n=======================\nIN NO EVENT WILL BROCADE OR ITS LICENSORS, SUPPLIERS, RESELLERS OR DISTRIBUTORS\nBE LIABLE FOR ANY LOST REVENUE, PROFIT, OR DATA, OR FOR SPECIAL, INDIRECT,\nCONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES HOWEVER CAUSED AND REGARDLESS\nOF THE THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE\nSOFTWARE EVEN IF BROCADE OR ITS LICENSORS, SUPPLIERS, RESELLERS OR DISTRIBUTORS\nHAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL BROCADE'S\nOR ITS LICENSORS', SUPPLIERS', RESELLERS' OR DISTRIBUTORS' TOTAL LIABILITY TO\nEND USER, WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, EXCEED\nTHE PRICE PAID BY END USER. THE FOREGOING LIMITATIONS SHALL APPLY EVEN IF THE\nABOVE-STATED WARRANTY FAILS OF ITS ESSENTIAL PURPOSE. BECAUSE SOME STATES OR\nJURISDICTIONS DO NOT ALLOW LIMITATION OR EXCLUSION OF CONSEQUENTIAL OR INCIDENTAL\nDAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO END USER.\n\nProprietary Rights\n==================\nThe Software, including all intellectual property rights therein, is and will\nremain the sole and exclusive property of Brocade or its licensors.\n\nTerm and Termination\n====================\nThis Agreement is effective until terminated. End User's license rights under this\nAgreement will terminate immediately without notice from Brocade if End User fails\nto comply with any provision of this Agreement. Upon termination, End User must stop\nusing the Software and destroy all copies of Software and the corresponding license\nkeys in its possession or control.\n\nCompliance with Law\n===================\nEach party agrees to comply with all applicable laws, rules and regulations in\nconnection with its activities under this Agreement. Without limiting the foregoing,\nEnd User acknowledges and agrees that the Software, including technical data, is \nsubject to United States export control laws, including the United States Export\nAdministration Act and its associated regulations, and may be subject to export or\nimport regulations in other countries. End User agrees to comply strictly with all\nsuch regulations and acknowledges that End User has the responsibility to obtain \nlicenses to export, re-export, or import the Software.\n\nRestricted Rights\n=================\nThe Software under this Agreement is commercial computer software as that term is\ndescribed in 48 C.F.R. 252.227-7014(a)(1). If acquired by or on behalf of a civilian\nagency, the U.S. Government acquires this commercial computer software and/or\ncommercial computer software documentation subject to the terms of this Agreement\nas specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data)\nof the Federal Acquisition Regulations (\"FAR\") and its successors. If acquired by\nor on behalf of any agency within the Department of Defense (\"DOD\"), the\nU.S. Government acquires this commercial computer software and/or commercial computer\nsoftware documentation subject to the terms of this Agreement as specified in\n48 C.F.R. 227.7202-3 of the DOD FAR Supplement (\"DFAR\") and its successors.\n\nGeneral\n=======\nThis Agreement will bind and inure to the benefit of each party's successors and\nassigns, provided that End User may not assign or transfer this Agreement, in whole\nor in part, without Brocade's written consent. This Agreement shall be governed by\nand construed in accordance with the laws of the State of California,\nUnited States of America, as if performed wholly within the state and without giving\neffect to the principles of conflict of law. No failure of either party to exercise\nor enforce any of its rights under this Agreement will act as a waiver of such\nrights. If any portion hereof is found to be void or unenforceable, the remaining\nprovisions of this Agreement shall remain in full force and effect. This Agreement\nis the complete and exclusive agreement between the

```
 parties with respect to the\nsubject matter hereof, superseding and replacing any and all prior agreements,
\ncommunications, and understandings (both written and oral) regarding such subject\nmatter.\n\nEffective
 October 1, 2019\n"
  }
 }
}
```


brocade-license/license

Description

A list of licenses installed on the switch.

Parameters

Table 94: brocade-license/license

Name	Type	Description
name	string	The representation of the license would be either license key or serial number. The license key is a string with alpha numeric characters and the License serial number is a string with the format of 'FOS-XX-X-XX-XXXXXXXX'. Example of a license key and serial number mentioned below. License key: 'HP9ttZNSgmB4MCD3NmNWgQDWtAKBFtXtBSFJF' Serial number: 'FOS-00-0-02-11201234'
capacity	uint32	The capacity for the license installed on the switch. This parameter is displayed only for a capacity-based license.
consumed	uint32	The number of slots configured to use the license installed on the switch. This leaf is displayed only for a slot-based license.
expiration-date	string	The expiration date for the license installed on the switch. The expiration date for the license in 'MM/DD/YYYY' format. This attribute is displayed only when the time based license installed on the system.
generation-date	string	The generation date for the license installed on the switch. The generation date for the license in 'MM/DD/YYYY' format. This leaf is valid only for serial number based licenses.
active-firmware-version	fru:firmware-version-type	A human readable string identifying the running firmware version for the Trusted FOS (TruFOS) Certificate license installed on a switch. Example of running firmware version mentioned below. '9.0.0','9.0.0+','9.0.0++','9.1.0-','9.0.0--','9.0.0*,9.1.0+'
target-firmware-version	fru:firmware-version-type	A human readable string identifying the target firmware version for the Trusted FOS (TruFOS) Certificate license installed on a switch. Example of target firmware version mentioned below. '9.0.0','9.0.0+','9.0.0++','9.1.0-','9.0.0--','9.0.0*,9.1.0+'
license-format	enumeration	To determine the specified license is certificate or key based.
features	Container	A list of features that are integrated in a single license key
- feature	string	The name of the feature.

Name	Type	Description
configured-blade-slots	Container	A list of slot numbers of the configured blade slots for the license installed on the switch.
- configured-blade-slot	uint32	The configured blade slot details. This leaf is displayed only for a slot based license.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <license>
  <name>QYRKQTWJNfCZHCf3NNfBgJLWYWCDrrLSB7BZN</name>
  <features>
  <feature>Extended Fabric</feature>
  <feature>Trunking</feature>
  <feature>FICON_CUP</feature>
  <feature>Fabric Vision and IO Insight</feature>
 </features>
  <license-format>key-based-license</license-format>
 </license>
 <license>
  <name>rCTRGSBSB9BBPZEtAYB4g7ZTZ3C974PQANXNCZFABHDB</name>
  <features>
   <feature>Q-Flex Ports on Demand</feature>
  </features>
  <capacity>32</capacity>
  <license-format>key-based-license</license-format>
 </license>
 <license>
  <name>AgP4QaZ3J7LNSKRDTfrg9GgLQD3A4DSfLHYf4THAfaAA</name>
  <features>
   <feature>Ports on Demand</feature>
  </features>
  <capacity>48</capacity>
  <license-format>key-based-license</license-format>
 </license>
 <license>
  <name>PPRgJXLGGXDQ9DHNrRK9HTCGS7HSAHDtMWSD4SGAZKFA</name>
  <features>
```

```
    <feature>Integrated Routing Ports on Demand</feature>
   </features>
   <capacity>128</capacity>
   <license-format>key-based-license</license-format>
  </license>
 </Response>
```

brocade-license/ports-on-demand-license-info

Description

Contains information related to ports-on-demand license reservations.

Parameters

Table 95: brocade-license/ports-on-demand-license-info

Name	Type	Description
sfp-port-count	yang:gauge32	Total count of SFP based ports available on the switch.
qsfp-port-count	yang:gauge32	Total count of QSFP based ports available on the switch.
sfp-dd-count	yang:gauge32	Total count of SFP Double Density (dd) based ports available on the switch.
port-on-demand-license	boolean	Specifies if ports on demand license is installed or not. true : Ports on demand license is installed. false : Ports on demand license is not installed.
qflex-port-on-demand-license	boolean	Specifies if Q-flex ports on demand license is installed. true : Q-flex ports on demand license installed. false : Q-flex ports on demand license is not installed.
sfp-dd-port-on-demand-license	boolean	Specifies if Double Density (dd) ports on demand license is installed. true : Double Density ports on demand license installed. false : Double Density ports on demand license is not installed.
provisioned-sfp-port-count	yang:gauge32	Total number of SFP based ports provisioned for use in the switch.
sfp-port-base-switch-allowance-count	yang:gauge32	Indicates port assignments provisioned by base switch allowance.
sfp-port-license-allowance-count	yang:gauge32	Indicates maximum assignments provisioned by SFP based ports on demand license.
sfp-port-assigned-count	yang:gauge32	Total number of SFP based ports assigned to base switch allowance or installed licenses.
sfp-port-base-switch-assigned-count	yang:gauge32	Indicates number of SFP ports assigned to base switch allowance.
licensed-sfp-port-count	yang:gauge32	Indicates number of SFP ports assigned to the ports on demand license.
sfp-port-license-reservation-available-count	yang:gauge32	Indicates license reservations available for use by unassigned SFP ports.
provisioned-qsfp-port-count	yang:gauge32	Total number of QSFP based ports provisioned for use in the switch.

Name	Type	Description
qsfp-base-switch-allowance-count	yang:gauge32	Indicates port assignments provisioned by base switch allowance.
qsfp-license-allowance-count	yang:gauge32	Indicates maximum assignments provisioned by QSFP ports on demand license.
qsfp-port-assigned-count	yang:gauge32	Total number of QSFP based ports assigned to base switch allowance or installed licenses.
qsfp-port-base-switch-assigned-count	yang:gauge32	Indicates number of QSFP ports assigned to base switch allowance
licensed-qsfp-port-count	yang:gauge32	Indicates number of QSFP ports assigned to the ports on demand license.
qsfp-port-license-reservation-available-count	yang:gauge32	Indicates license reservations available for use by unassigned QSFP ports.
provisioned-sfp-dd-port-count	yang:gauge32	Total number of SFP-DD based ports provisioned for use in the switch.
sfp-dd-base-switch-allowance-count	yang:gauge32	Indicates port assignments provisioned by base switch allowance.
sfp-dd-license-allowance-count	yang:gauge32	Indicates maximum assignments provisioned by SFP-DD ports on demand license.
sfp-dd-port-assigned-count	yang:gauge32	Total number of SFP-DD based ports assigned to base switch allowance or installed licenses.
sfp-dd-port-base-switch-assigned-count	yang:gauge32	Indicates number of SFP-DD ports assigned to base switch allowance
licensed-sfp-dd-port-count	yang:gauge32	Indicates number of SFP-DD ports assigned to the ports on demand license.
sfp-dd-port-license-reservation-available-count	yang:gauge32	Indicates license reservations available for use by unassigned SFP-DD ports.
sfp-port-base-switch-members	Container	Contains list of SFP based ports assigned to base switch allowance.
- port	fibrechannel:port-number-type	SFP based port assigned to base switch allowance.
licensed-sfp-port-members	Container	Contains list of SFP ports assigned to ports-on-demand license.
- port	fibrechannel:port-number-type	SFP based port assigned to port on demand license.
unassigned-sfp-port-members	Container	Contains list of SFP ports not assigned to base switch or ports on demand license.
- port	fibrechannel:port-number-type	SFP based port not assigned to base switch or port on demand license.
licensed-offline-sfp-port-members	Container	Contains list of licensed SFP ports that are offline.
- port	fibrechannel:port-number-type	Licensed SFP port that is offline.
qsfp-port-base-switch-members	Container	Contains list of QSFP based ports assigned to base switch allowance.

Name	Type	Description
- port	fibrechannel:port-number-type	QSFP based port assigned to base switch allowance.
licensed-qsfp-port-members	Container	Contains list of QSFP ports assigned to port on demand license.
- port	fibrechannel:port-number-type	QSFP based port assigned to port on demand license.
unassigned-qsfp-port-members	Container	Contains list of QSFP ports not assigned to base switch or ports on demand license.
- port	fibrechannel:port-number-type	QSFP based port member not assigned to base switch or ports on demand license.
licensed-offline-qsfp-port-members	Container	Contains list of licensed QSFP ports that are offline.
- port	fibrechannel:port-number-type	Licensed QSFP port that is offline.
sfp-dd-port-base-switch-members	Container	Contains list of SFP-DD based ports assigned to base switch allowance.
- port	fibrechannel:port-number-type	SFP-DD based port member assigned to base switch allowance.
licensed-sfp-dd-port-members	Container	Contains list of SFP-DD ports assigned to ports-on-demand license.
- port	fibrechannel:port-number-type	SFP-DD based port member assigned to ports-on-demand license.
unassigned-sfp-dd-port-members	Container	Contains list of SFP-DD ports not assigned to base switch or ports on demand license.
- port	fibrechannel:port-number-type	SFP-DD based port member not assigned to base switch or ports on demand license.
licensed-offline-sfp-dd-port-members	Container	Contains list of licensed SFP-DD ports that are offline.
- port	fibrechannel:port-number-type	Licensed SFP-DD port that is offline.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <ports-on-demand-license-info>
  <sfp-port-count>96</sfp-port-count>
  <qsfp-port-count>32</qsfp-port-count>
  <sfp-dd-count>0</sfp-dd-count>
  <port-on-demand-license>true</port-on-demand-license>
  <qflex-port-on-demand-license>true</qflex-port-on-demand-license>
  <sfp-dd-port-on-demand-license>false</sfp-dd-port-on-demand-license>
  <provisioned-sfp-port-count>96</provisioned-sfp-port-count>
  <sfp-port-base-switch-allowance-count>48</sfp-port-base-switch-allowance-count>
  <sfp-port-license-allowance-count>48</sfp-port-license-allowance-count>
  <sfp-port-assigned-count>96</sfp-port-assigned-count>
  <sfp-port-base-switch-assigned-count>48</sfp-port-base-switch-assigned-count>
  <licensed-sfp-port-count>48</licensed-sfp-port-count>
  <sfp-port-license-reservation-available-count>0</sfp-port-license-reservation-available-count>
  <sfp-port-base-switch-members>
   <sfp-port-base-switch-members>0</sfp-port-base-switch-members>
   <sfp-port-base-switch-members>1</sfp-port-base-switch-members>
   <sfp-port-base-switch-members>2</sfp-port-base-switch-members>
   <sfp-port-base-switch-members>3</sfp-port-base-switch-members>
...
   <sfp-port-base-switch-members>93</sfp-port-base-switch-members>
   <sfp-port-base-switch-members>94</sfp-port-base-switch-members>
   <sfp-port-base-switch-members>95</sfp-port-base-switch-members>
  </sfp-port-base-switch-members>
   <licensed-sfp-port-members>
   <licensed-sfp-port-members>10</licensed-sfp-port-members>
   <licensed-sfp-port-members>11</licensed-sfp-port-members>
   <licensed-sfp-port-members>12</licensed-sfp-port-members>
...
   <licensed-sfp-port-members>58</licensed-sfp-port-members>
   <licensed-sfp-port-members>59</licensed-sfp-port-members>
   <licensed-sfp-port-members>61</licensed-sfp-port-members>
   <licensed-sfp-port-members>63</licensed-sfp-port-members>
  </licensed-sfp-port-members>
  <unassigned-sfp-port-members/>
  <licensed-offline-sfp-port-members>
   <licensed-offline-sfp-port-members>0</licensed-offline-sfp-port-members>
   <licensed-offline-sfp-port-members>1</licensed-offline-sfp-port-members>
   <licensed-offline-sfp-port-members>2</licensed-offline-sfp-port-members>
...
   <licensed-offline-sfp-port-members>93</licensed-offline-sfp-port-members>
   <licensed-offline-sfp-port-members>94</licensed-offline-sfp-port-members>
   <licensed-offline-sfp-port-members>95</licensed-offline-sfp-port-members>
  </licensed-offline-sfp-port-members>
  <provisioned-qsfp-port-count>32</provisioned-qsfp-port-count>
  <qsfp-base-switch-allowance-count>0</qsfp-base-switch-allowance-count>
  <qsfp-license-allowance-count>32</qsfp-license-allowance-count>
  <qsfp-port-assigned-count>32</qsfp-port-assigned-count>
  <qsfp-port-base-switch-assigned-count>0</qsfp-port-base-switch-assigned-count>
  <licensed-qsfp-port-count>32</licensed-qsfp-port-count>
  <qsfp-port-license-reservation-available-count>0</qsfp-port-license-reservation-available-count>
```

```
   <qsfp-port-base-switch-members/>
   <licensed-qsfp-port-members>
    <licensed-qsfp-port-members>96</licensed-qsfp-port-members>
    <licensed-qsfp-port-members>97</licensed-qsfp-port-members>
    <licensed-qsfp-port-members>98</licensed-qsfp-port-members>
 ...
    <licensed-qsfp-port-members>125</licensed-qsfp-port-members>
    <licensed-qsfp-port-members>126</licensed-qsfp-port-members>
    <licensed-qsfp-port-members>127</licensed-qsfp-port-members>
   </licensed-qsfp-port-members>
   <unassigned-qsfp-port-members/>
   <licensed-offline-qsfp-port-members>
    <licensed-offline-qsfp-port-members>96</licensed-offline-qsfp-port-members>
    <licensed-offline-qsfp-port-members>97</licensed-offline-qsfp-port-members>
    <licensed-offline-qsfp-port-members>98</licensed-offline-qsfp-port-members>
 ...
    <licensed-offline-qsfp-port-members>125</licensed-offline-qsfp-port-members>
    <licensed-offline-qsfp-port-members>126</licensed-offline-qsfp-port-members>
    <licensed-offline-qsfp-port-members>127</licensed-offline-qsfp-port-members>
   </licensed-offline-qsfp-port-members>
   <provisioned-sfp-dd-port-count>0</provisioned-sfp-dd-port-count>
   <sfp-dd-base-switch-allowance-count>0</sfp-dd-base-switch-allowance-count>
   <sfp-dd-license-allowance-count>0</sfp-dd-license-allowance-count>
   <sfp-dd-port-assigned-count>0</sfp-dd-port-assigned-count>
   <sfp-dd-port-base-switch-assigned-count>0</sfp-dd-port-base-switch-assigned-count>
   <licensed-sfp-dd-port-count>0</licensed-sfp-dd-port-count>
   <sfp-dd-port-license-reservation-available-count>0</sfp-dd-port-license-reservation-available-count>
   <sfp-dd-port-base-switch-members/>
   <licensed-sfp-dd-port-members/>
   <unassigned-sfp-dd-port-members/>
   <licensed-offline-sfp-dd-port-members/>
  </ports-on-demand-license-info>
 </Response>
```

brocade-lldp/lldp-global

Description

The LLDP switch level configuration and operational parameters.

Parameters

Table 96: brocade-lldp/lldp-global

Name	Type	Description
multiplier	uint32	The LLDP timeout multiplier value. Default value is 4
tx-interval	uint32	The LLDP Tx interval of the switch. Default value is 30 seconds
system-name	string	The LLDP system name of the switch.

Name	Type	Description
system-description	string	The LLDP system description of the switch.
enabled-state	boolean	The LLDP protocol state of the switch. The possible values are: true : LLDP is enabled on the switch false : LLDP is disabled on the switch Default value is true.
mandatory-tlvs	Container	The mandatory LLDP TLVs supported by the switch.
- tlv	lldp-mandatory-tlv-name-type	The list of mandatory LLDP TLVs (chassis-id, port-id and time-to-live) which cannot be modified.
optional-tlvs	Container	The optional LLDP TLVs supported by the switch.
- tlv	lldp-optional-tlv-name-type	The list of optional TLVs enabled on the switch. The dcbx, fcoe-app, fcoe-lls, sys-name and port-desc TLVs are enabled by default and user can disable them if required. The dcbx TLV should be enabled beforehand to enable fcoe-app and fcoe-lls TLVs.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <lldp-global>
  <multiplier>4</multiplier>
  <tx-interval>30</tx-interval>
  <system-name>Switch_Test</system-name>
  <system-description>Brocade_X7_4_Fabric_OS_Version_9_0_0</system-description>
  <enabled-state>true</enabled-state>
  <mandatory-tlvs>
   <tlv>chassis-id</tlv>
   <tlv>port-id</tlv>
   <tlv>time-to-live</tlv>
  </mandatory-tlvs>
  <optional-tlvs>
   <tlv>port-desc</tlv>
   <tlv>sys-name</tlv>
   <tlv>dcbx</tlv>
   <tlv>fcoe-app</tlv>
   <tlv>fcoe-lls</tlv>
```

```
   </optional-tlvs>
  </lldp-global>
 </Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-lldp/lldp-neighbor

Description

The list of LLDP neighbor devices connected to the switch.

Parameters

Table 97: brocade-lldp/lldp-neighbor

Name	Type	Description
dead-interval	uint32	The dead interval of the LLDP neighbor.
remaining-life	uint32	The remaining life of the LLDP neighbor.
remote-interface-name	string	The remote interface name of the LLDP neighbor device connected to local switch ethernet interface.
chassis-id	string	The chassis ID of the LLDP neighbor.
system-name	string	The system name of the LLDP neighbor.
slot-port	fibrechannel:slot-port-name-type	The local interface name.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <lldp-neighbor>
  <slot-port>8/2</slot-port>
  <dead-interval>120</dead-interval>
  <remaining-life>102</remaining-life>
  <remote-interface-name>ge2</remote-interface-name>
  <chassis-id>c4f5.7cdc.c0fa</chassis-id>
  <system-name>B7840_073_017</system-name>
 </lldp-neighbor>
</Response>
```

brocade-lldp/lldp-profile

Description

The LLDP profile.

Parameters

Table 98: brocade-lldp/lldp-profile

Name	Type	Description
name	string	The LLDP profile name.
multiplier	int32	The LLDP timeout multiplier value of the LLDP profile. Multiplier value -1 indicates that it will be derived from lldp global multiplier value.
tx-interval	int32	The Transmit interval value of the LLDP profile. Tx-interval value -1 indicates that it will be derived from lldp global tx-interval value.
enabled-tlvs	Container	The enabled TLVs for the LLDP profile.
- tlv	lldp-optional-tlv-name-type	List of enabled TLVs for the LLDP profile. Empty enabled-tlvs indicates that the tlvs will be derived from lldp global optional-tlvs.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-lldp/lldp-statistics

Description

The LLDP protocol specific Tx, Rx and Error statistics

of the port/s.

Parameters

Table 99: brocade-lldp/lldp-statistics

Name	Type	Description
out-frames	ietf:zero-based-counter64	The number of LLDP out frames.
in-frames	ietf:zero-based-counter64	The number of LLDP In frames.
frames-aged-out	ietf:zero-based-counter64	The number of LLDP aged out frames.
frames-discarded	ietf:zero-based-counter64	The number of LLDP discarded frames.
error-frames	ietf:zero-based-counter64	The number of LLDP error frames
tlv-discarded	ietf:zero-based-counter64	The number of discarded LLDP TLVs.
tlv-unrecognized	ietf:zero-based-counter64	The number of unrecognized LLDP TLVs.
slot-port	fibrechannel:slot-port-name-type	Port name.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
 <?xml version="1.0"?>
<Response>
 <lldp-statistics>
  <slot-port>4/1</slot-port>
  <out-frames>0</out-frames>
  <frames-aged-out>0</frames-aged-out>
  <frames-discarded>0</frames-discarded>
  <error-frames>0</error-frames>
  <in-frames>0</in-frames>
  <tlv-discarded>0</tlv-discarded>
  <tlv-unrecognized>0</tlv-unrecognized>
 </lldp-statistics>
</Response>
```

brocade-logging/audit

Description

The audit logging configuration parameters. You can configure certain filter classes, to set severity levels for audit messages, and enable or disable audit filters. Depending on the configuration, certain classes are logged to syslog for auditing. Note that syslog configuration is required for logging audit messages.

Parameters

Table 100: brocade-logging/audit

Name	Type	Description
audit-enabled	boolean	Enables or disables the audit filters. Note that this does not change the log class and log severity configuration. true : The audit filter is enabled. false : The audit filter is disabled.
severity-level	brocade-logging-types:severity-level-type	The severity level of the log messages you want to display. Displays log messages of the specified severity level and higher. You cannot enter multiple severity levels.
filter-class-list	Container	A list of class types needed for audit configuration.
- filter-class	brocade-logging-types:class-type	The filters to be configured for audit classes. You can set more than one audit class, separated by a comma.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <audit>
  <audit-enabled>true</audit-enabled>
  <severity-level>info</severity-level>
  <filter-class-list>
   <filter-class>zone</filter-class>
   <filter-class>security</filter-class>
   <filter-class>configuration</filter-class>
   <filter-class>firmware</filter-class>
   <filter-class>fabric</filter-class>
   <filter-class>ls</filter-class>
   <filter-class>cli</filter-class>
   <filter-class>maps</filter-class>
  </filter-class-list>
 </audit>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-logging/audit-log

Description

This is used for listing all the auditlog events generated on the switch

and accumulated over a period of time.

Parameters

Table 101: brocade-logging/audit-log

Name	Type	Description
sequence-number	uint32	Sequence number to identify the generated auditlog message.
time-stamp	yang:date-and-time	Timestamp of the auditlog event generated.
message-id	brocade-logging-types:message-id-type	It uniquely identifies an auditlog message.
switch-user-friendly-name	string	A printable ASCII name for the switch or chassis based on the context of the auditlog message.

Name	Type	Description
message-text	string	Message text populated with actual values in the auditlog message.
severity-level	brocade-logging-types:severity-level-type	Severity level of the auditlog message.
event-class	brocade-logging-types:event-class-type	Event class of the auditlog message.
ip-address	union {inet:host;string;}	IP-address from which user is logged in to the switch.
user-name	union {brocade-security-type:user-config-login-user-name-type;string;}	User at the switch session which generated the auditlog, None if the generated audilog doesn't have a user at the specific context.
role	brocade-security-type:user-config-role-type	The user's role for the logical fabric context where the auditlog was generated.
interface	fibrechannel:interface-type	Interface used to login to the switch which generated the auditlog.
application-name	string	Name of the Switch Manageability application run by user.
fabric-id	uint32	Fabric id of switch that generated the auditlog. It is 0 : in chassis context 1-128 : fabric id of the logical switch in switch context.
event-info	string	This leaf may contain additional identifying information to help isolate the event, when applicable. For example, slot and port values in slot/port format.
application-user-name	string	Name of the Application User Name provided by user. None: Displays, if the application user name is not configured NA: Displays, when not applicable.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <audit-log>
  <sequence-number>1</sequence-number>
  <time-stamp>2020-03-07T10:40:31-07:00</time-stamp>
  <message-id>RAS-3005</message-id>
  <message-text>CLI: ## REST Operation: GET /brocade-maps/rule/name/defALL_32GSWL_SFPSFP_TEMP_85/actions ##</
message-text>
```

```
   <severity-level>info</severity-level>
   <event-class>cli</event-class>
   <user-name>root</user-name>
   <role>root</role>
   <ip-address>10.38.255.47</ip-address>
   <interface>ssh</interface>
   <application-name>CLI</application-name>
   <switch-user-friendly-name>BrocadeX6-4</switch-user-friendly-name>
   <fabric-id>0</fabric-id>
  </audit-log>
 ...
  <audit-log>
   <sequence-number>8192</sequence-number>
   <time-stamp>2020-03-09T14:55:11-06:00</time-stamp>
   <message-id>SEC-3022</message-id>
   <message-text>Event: logout, Status: success, Info: Successful logout by user [admin].</message-text>
   <severity-level>info</severity-level>
   <event-class>security</event-class>
   <user-name>admin</user-name>
   <role>admin</role>
   <ip-address>2620:100:4:f800:215:5dff:fe26:3e0c</ip-address>
   <interface>https</interface>
   <application-name>SANnavMP-2.1.0-f</application-name>
   <switch-user-friendly-name>66-243_X6-4_Venator</switch-user-friendly-name>
   <fabric-id>128</fabric-id>
  </audit-log>
 </Response>
```

brocade-logging/error-log

Description

This is used for listing all the raslog events generated on a switch and accumulated over a period of time.

Parameters

Table 102: brocade-logging/error-log

Name	Type	Description
sequence-number	uint32	Externally assigned sequence number to identify the generated raslog message.
time-stamp	yang:date-and-time	Timestamp of the raslog event generated.
message-id	brocade-logging-types:message-id-type	It uniquely identifies a raslog message.
fabric-id	uint32	Fabric id of switch that generated the raslog. It is 0 : in chassis context 1-128 : fabric id of the logical switch in switch context

Name	Type	Description
slot-id	uint32	It denotes slot id of the cp that generated raslog. It is zero for pizza-box systems.
severity-level	brocade-logging-types:severity-level-type	Severity level of the log message.
ffdc-generated-event	boolean	Whether generated raslog is an FFDC event or not. The values denote, true : The raslog is an FFDC event, false : The raslog is not an FFDC event.
switch-user-friendly-name	string	A printable ASCII name for the switch or chassis based on the context of the raslog message.
message-text	string	Message text populated with actual values in the raslog.
event-info	string	This leaf may contain additional identifying information to help isolate the event, when applicable. For example, slot and port values in slot/port format.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <error-log>
  <time-stamp>2020-03-02T10:01:37-07:00</time-stamp>
  <message-id>HAM-1004</message-id>
  <sequence-number>1</sequence-number>
  <severity-level>info</severity-level>
  <ffdc-generated-event>false</ffdc-generated-event>
  <fabric-id>0</fabric-id>
  <slot-id>1</slot-id>
  <switch-user-friendly-name>BrocadeX6-4</switch-user-friendly-name>
  <message-text>Processor rebooted - Reset.</message-text>
 </error-log>
...
 <error-log>
  <time-stamp>2020-03-09T14:42:56-06:00</time-stamp>
  <message-id>MAPS-1132</message-id>
  <sequence-number>723</sequence-number>
  <severity-level>info</severity-level>
  <ffdc-generated-event>false</ffdc-generated-event>
```

```
   <fabric-id>128</fabric-id>
   <slot-id>1</slot-id>
   <switch-user-friendly-name>66-243_X6-4_Venator</switch-user-friendly-name>
   <message-text>Monitoring on members 8/16 of type PORT is restarted.</message-text>
  </error-log>
 </Response>
```

brocade-logging/log-quiet-control

Description

A list of parameters that control quiet time for logging based on the log type.

Parameters

Table 103: brocade-logging/log-quiet-control

Name	Type	Description
log-type	string	The log type for which quiet time is configured. Valid values are audit and raslog.
quiet-enabled	boolean	Whether quiet time is enabled for logs. true : Quiet time is enabled. false : Quiet time is disabled.
start-time	brocade-logging-types:time-24hr-type	The RAS quiet start time for the specified log type. The quiet start time in the 24 hour clock format hh:mm. hh : Hours [00 to 23] mm : Minutes [00 to 59]
end-time	brocade-logging-types:time-24hr-type	The RAS quiet end time for the specified log type. Start time must be configured before you can set the end time. The quiet start time in the 24 hour clock format hh:mm. hh : Hours [00 to 23] mm : Minutes [00 to 59]
days-of-week	Container	The days of the week for which you want to set quiet time for the specified log type.
- day	string	The days of the week for which you want to set quiet time for the specified log type. You can use this along with the start and end times. Empty string : Disables quiet time for the specified log. everyday : Enables quiet time everyday for the specified start time and end time. forever : Enables quiet time every day all day (you cannot enter a start or end time). [mon..sun] : Enables quiet time for the specified days of the week for the specified start time and end time.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <log-quiet-control>
  <log-type>audit</log-type>
  <quiet-enabled>false</quiet-enabled>
  <days-of-week/>
 </log-quiet-control>
 <log-quiet-control>
  <log-type>raslog</log-type>
  <quiet-enabled>false</quiet-enabled>
  <days-of-week/>
 </log-quiet-control>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-logging/log-setting

Description

The system-wide parameters to control the logs.

Parameters

Table 104: brocade-logging/log-setting

Name	Type	Description
syslog-facility-level	string	The facility level determines the priority of the syslog messages being recorded at the server. A smaller facility level corresponds to higher priority syslog messages.
syslog-source-ip-chassis	boolean	Enable or disable Syslog messages source IP as chassis IP. true : Set syslog source ip as chassis IP false : Set syslog source ip as CP IP.

Name	Type	Description
keep-alive-period	uint16	The RASlog keep alive timeout. The system default is 1. 0 : Disables the keep alive timeout. [1..24] : Keeps logging alive for the specified time.
clear-log	enumeration	Operation attribute to select the log to be cleared. Values can be error-log or audit-log.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "log-setting": {
            "syslog-facility-level": "log_local7",
            "keep-alive-period": 1
        }
    }
}
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-logging/management-session-login-information

Description

Lists all the login sessions information including

history and active sessions.

This list supports the following query parameter combinations:

active

Parameters

Table 105: brocade-logging/management-session-login-information

Name	Type	Description
timestamp	yang:date-and-time	The login timestamp.
host	inet:host	The client IP address or domain name.
user-name	string	The user name configured on the server.
application-name	string	The client application name input request http header – User-Agent.
session-id	string	The client session ID.
vf-id	fibrechannel:fabric-id-type	This is the virtual fabric identification (VFID) of the logical switch.
auth-token-present	boolean	It indicates that whether the auth-token is generated for this user or not. true : Auth-token is available for this user. false: Auth-token is not generated for this user.
protocol	enumeration	Protocol used to communicate with Management applications.
active	boolean	The active-session option indicates that the management session is active or history of logins. An active session is also present in the history of logins. true : Displays only active sessions. false: Displays history of session logins, including active sessions.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <management-session-login-information>
        <timestamp>2021-11-05T14:24:58-06:00</timestamp>
        <host>10.10.10.24</host>
        <user-name>admin</user-name>
        <application-name>python-requests/2.23.0</application-name>
        <session-id>0a673e86dc08c73bbe2efeda0b1244713927f77383f47bb35f9b426ab04728db</session-id>
        <auth-token-present>false</auth-token-present>
        <protocol>https</protocol>
        <active>false</active>
```

```
    </management-session-login-information>
...
    <management-session-login-information>
        <timestamp>2021-11-08T12:46:03-07:00</timestamp>
        <host>10.10.10.94</host>
        <user-name>admin</user-name>
        <application-name>PostmanRuntime/7.28.4</application-name>
        <session-id>69b6accfa60c33e9ee832c68dc1351e25bcc21906b126f9a3b4d7a6a08df8722</session-id>
        <auth-token-present>false</auth-token-present>
        <protocol>https</protocol>
        <active>true</active>
    </management-session-login-information>
</Response>
```

brocade-logging/raslog

Description

A list of RASlog messages' configurable parameters, such as enabling message logging, the message ID, the current severity level, transferring to the syslog server is enabled. This list also includes read-only leafs, such as the default severity, the message text, and the flooded flag. A maximum of 50 RASlog objects can be modified in a single PATCH request.

Parameters

Table 106: brocade-logging/raslog

Name	Type	Description
message-id	brocade-logging-types:message-id-type	The unique identifier for each RASlog message.
message-enabled	boolean	Whether message logging is enabled RASlog message identified by message ID. true : Message logging is enabled. false : Message logging is disabled.
message-flooded	boolean	Whether the RASlog message is blocked due to flooding. true : The message is blocked due to flooding. false : The message is not blocked due to flooding.
syslog-enabled	boolean	Whether internal RASlog messages to be sent to syslog is enabled. true : Syslog is enabled. false : Syslog is disabled.
message-text	brocade-logging-types:ascii-text-type	The message text corresponding to the RASlog message.

Name	Type	Description
current-severity	union {brocade-logging-types:severity-level-type;string;}	The current severity level of the RASlog. The default security level sets severity to a pre-defined level of logging for the RASlog associated with the corresponding message ID. However, you can configure a different severity with this parameter. Valid values are info, warning, error, critical and default.
default-severity	brocade-logging-types:severity-level-type	The pre-defined default severity level of the RASlog. Values displayed are info, warning, error and critical.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <raslog>
  <message-id>FCIP-1000</message-id>
  <current-severity>error</current-severity>
  <default-severity>error</default-severity>
  <message-enabled>true</message-enabled>
  <message-flooded>false</message-flooded>
 </raslog>

...

 <raslog>
  <message-id>CONF-1055</message-id>
  <current-severity>info</current-severity>
  <default-severity>info</default-severity>
  <message-enabled>true</message-enabled>
  <message-flooded>false</message-flooded>
 </raslog>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-logging/raslog-module

Description

A list of Fabric OS modules (identified by module ID) that have logging enabled. You can use this list to enable logging for Fabric OS modules.

Parameters

Table 107: brocade-logging/raslog-module

Name	Type	Description
module-id	brocade-logging-types:module-id-type	The Fabric OS module identified by module ID.
log-enabled	boolean	Whether logging is enabled for the Fabric OS module. This parameter is used only during a PATCH request. Note that this parameter does not display when you use a GET request to display RASlog modules. true : Logging is enabled for the Fabric OS module. false : Logging is disabled for the Fabric OS module.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <raslog-module>
  <module-id>UT</module-id>
 </raslog-module>
...
 <raslog-module>
  <module-id>BCMG</module-id>
 </raslog-module>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-logging/supportftp

Description

To configure the server details at switch to transfer the debug logs
automatically when FFDC, panic or manual tracedumps are generated.

Parameters

Table 108: brocade-logging/supportftp

Name	Type	Description
host	union {inet:host;fibrechannel:brocade-empty-string-type;}	IP address or DNS name of server to which debug logs have to be send.
user-name	string	User name of account in server.
password	union {string;brocade-security-type:base64-string-type;}	Base64 encoded password for the account in server.
remote-directory	string	Directory path in server where switch log will be saved.
auto-enabled	boolean	Enable or disable auto dump of trace, FFDC, and core files to specified server. true : Enable auto dump. false : Disable auto dump.
protocol	union {brocade-security-type:sec-crypto-cfg-file-transfer-protocol-type;fibrechannel:brocade-empty-string-type;}	Specifies the protocol used to transfer the trace data as scp, ftp or sftp.
port	uint32	User defined port number for SCP (Secure Copy protocol) or SFTP (Secure File Transfer Protocol).
connectivity-check-interval	uint32	Specifies the time interval in units of hours at which the server connectivity is tested periodically. When set to 0, this connectivity test is disabled.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
 <?xml version="1.0"?>
<Response>
 <supportftp>
  <host/>
  <user-name/>
  <password/>
  <remote-directory/>
  <auto-enabled>false</auto-enabled>
  <protocol>ftp</protocol>
  <connectivity-check-interval>0</connectivity-check-interval>
 </supportftp>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-logging/syslog-server

Description

The remote syslog server. You can configure a switch to forward all error log entries to a remote syslog server. Brocade devices use the syslog daemon, a process available on most UNIX systems that reads and forwards system messages to the appropriate log files or users, depending on the system configuration.

Parameters

Table 109: brocade-logging/syslog-server

Name	Type	Description
server	inet:host	The IPv4 or IPv6 address or DNS name of the server.
port	inet:port-number	The target syslog server's port number. The default port number is 514 for non-secure mode and 6514 for secure mode.
secure-mode	boolean	Enables or disables secure syslog mode to send the error log messages securely using the TLS protocol to the syslog server. Secure syslog mode is disabled by default. true : Secure mode is enabled. false : Secure mode is disabled.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <syslog-server>
  <server>10.10.10.11</server>
  <port>514</port>
  <secure-mode>false</secure-mode>
 </syslog-server>
 <syslog-server>
  <server>10.10.10.12</server>
  <port>514</port>
  <secure-mode>false</secure-mode>
 </syslog-server>
 <syslog-server>
  <server>10.10.10.13</server>
  <port>514</port>
  <secure-mode>false</secure-mode>
 </syslog-server>
</Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-management-ip-interface/management-interface-lldp-neighbor

Description

The list of LLDP neighbor devices connected to

phsyical management interface

Parameters

Table 110: brocade-management-ip-interface/management-interface-lldp-neighbor

Name	Type	Description
dead-interval	uint32	The dead interval of the LLDP neighbor.
remaining-life	uint32	The remaining life of the LLDP neighbor.
remote-interface-name	string	The remote interface name of the LLDP neighbor device connected to local switch ethernet interface.
chassis-id	string	The chassis ID of the LLDP neighbor.
system-name	string	The system name of the LLDP neighbor.
cp-name	fibrechannel:cp-name-type	For fixed-port switch, valid name is cp0. For chassis switch, valid names are cp0 and cp1.
physical-interface	fibrechannel:physical-interface-type	Physical management interface name. For fixed-port switch, valid name is eth0. For chassis switch, valid names are eth0 and eth3.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <management-interface-lldp-neighbor>
        <cp-name>cp0</cp-name>
        <physical-interface>eth0</physical-interface>
        <dead-interval>120</dead-interval>
        <remaining-life>116</remaining-life>
        <remote-interface-name>Gi1/0/31</remote-interface-name>
        <chassis-id>00fc.ba07.c880</chassis-id>
        <system-name>BRM-3D2-C06-RS1.nw-am.broadcom.net</system-name>
    </management-interface-lldp-neighbor>
    <management-interface-lldp-neighbor>
        <cp-name>cp1</cp-name>
        <physical-interface>eth0</physical-interface>
        <dead-interval>120</dead-interval>
        <remaining-life>104</remaining-life>
        <remote-interface-name>Gi1/0/2</remote-interface-name>
        <chassis-id>00fc.ba07.c880</chassis-id>
```

```
        <system-name>BRM-3D2-C06-RS1.nw-am.broadcom.net</system-name>
    </management-interface-lldp-neighbor>
</Response>
```

brocade-management-ip-interface/management-interface-lldp-statistics

Description

The LLDP protocol specific Tx, Rx and Error statistics

of the phsyical management interface

Parameters

Table 111: brocade-management-ip-interface/management-interface-lldp-statistics

Name	Type	Description
out-frames	ietf:zero-based-counter64	The number of LLDP out frames.
in-frames	ietf:zero-based-counter64	The number of LLDP In frames.
frames-aged-out	ietf:zero-based-counter64	The number of LLDP aged out frames.
frames-discarded	ietf:zero-based-counter64	The number of LLDP discarded frames.
error-frames	ietf:zero-based-counter64	The number of LLDP error frames
tlv-discarded	ietf:zero-based-counter64	The number of discarded LLDP TLVs.
tlv-unrecognized	ietf:zero-based-counter64	The number of unrecognized LLDP TLVs.
cp-name	fibrechannel:cp-name-type	For fixed-port switch, valid name is cp0. For chassis switch, valid names are cp0 and cp1.
physical-interface	fibrechannel:physical-interface-type	Physical management iterface name. For fixed-port switch, valid name is eth0. For chassis switch, valid names are eth0 and eth3.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <management-interface-lldp-statistics>
```

```
        <cp-name>cp0</cp-name>
        <physical-interface>eth0</physical-interface>
        <out-frames>9253</out-frames>
        <frames-aged-out>0</frames-aged-out>
        <frames-discarded>0</frames-discarded>
        <error-frames>0</error-frames>
        <in-frames>9300</in-frames>
        <tlv-discarded>0</tlv-discarded>
        <tlv-unrecognized>0</tlv-unrecognized>
    </management-interface-lldp-statistics>
    <management-interface-lldp-statistics>
        <cp-name>cp1</cp-name>
        <physical-interface>eth0</physical-interface>
        <out-frames>9072</out-frames>
        <frames-aged-out>0</frames-aged-out>
        <frames-discarded>0</frames-discarded>
        <error-frames>0</error-frames>
        <in-frames>9118</in-frames>
        <tlv-discarded>0</tlv-discarded>
        <tlv-unrecognized>0</tlv-unrecognized>
    </management-interface-lldp-statistics>
</Response>
```

brocade-management-ip-interface/management-ip-interface

Description

Chassis level IP configurations for management interface of switch.

Parameters

Table 112: brocade-management-ip-interface/management-ip-interface

Name	Type	Description
dhcp-gateway	union {inet:ipv4-address;fibrechannel:brocade-empty-string-type;}	The IPv4 gateway of switch IP network obtained from DHCP server. It is valid only when management-ip-configurations includes 'dhcp'.
static-subnet-mask	union {inet:ipv4-address;fibrechannel:brocade-empty-string-type;}	Statically configured IPv4 subnet mask of the switch IP network. It is valid only when management-ip-configurations does not include 'dhcp'.
dhcp-subnet-mask	union {inet:ipv4-address;fibrechannel:brocade-empty-string-type;}	The IPv4 subnet mask of the switch IP network obtained from DHCP server. It is valid only when management-ip-configurations includes 'dhcp'.
dhcp-enabled	boolean	Whether DHCP configuration is enabled or not. DHCP(Dynamic Host Configuration Protocol) in IPv4 features DHCP address, mask, gateway and DNS configuration from DHCP server. true: DHCP configuration has to be set to or is in enabled state, false: DHCP configuration has to be set to or is in disabled state.

Name	Type	Description
dhcpv6-enabled	boolean	Whether DHCPv6 configuration is enabled or not. Dynamic Host Configuration Protocol in IPv6 features DHCPv6 address, stateless gateways and DNS configuration from DHCPv6 server. true: DHCPv6 configuration has to be set to or is in enabled state, false: DHCPv6 configuration has to be set to or is in disabled state.
ipv6-ddns-enabled	boolean	Whether DDNS configuration is enabled or not. DDNS(Dynamic Domain Name System) in IPv6 features host name, host address, mask, gateway and DNS configuration to DNS server. true: DDNS configuration has to be set to or is in enabled state, false: DDNS configuration has to be unset to or is in disabled state.
ipv6-auto-enabled	boolean	Whether IPv6 Stateless Auto configuration is enabled or not. Stateless Auto configuration in IPv6 featuring stateless addresses, stateless gateways and DNS configuration from DHCPv6 server. true: Auto configuration has to be set to or is in enabled state, false: Auto configuration has to be set to or is in disabled state.
static-ip-addresses	Container	List of out-of-band static IPv4 and IPv6 addresses assigned to the ethernet port of the switch.
- ip-address	union {inet:ip-address;inet:ip-prefix;}	List of out-of-band static IP addresses assigned to the ethernet port of the switch.
stateful-ip-addresses	Container	List of out-of-band DHCP and DHCPv6 addresses assigned to the ethernet port of the switch.
- ip-address	union {inet:ip-address;inet:ip-prefix;}	List of out-of-band stateful IP addresses assigned to the ethernet port of the switch.
stateless-ipv6-addresses	Container	List of out-of-band stateless IPv6 addresses assigned to the ethernet port of the switch.
- ip-address	union {inet:ipv6-address;inet:ipv6-prefix;}	List of out-of-band stateless IP addresses assigned to the ethernet port of the switch.
static-ip-gateways	Container	List of IPv4 and IPv6 static gateway addresses for the switch IP address.
- ip-gateway	fru:ip-gateway-type	List of static gateway IP addresses of an IP router that can route packets to the destination IP address. The gateway address must be on the same IP subnet as one of the port IP addresses. Only one IPv4 static gateway and one IPv6 static gateway are allowed.
stateless-ipv6-gateways	Container	List of IPv6 stateless gateway addresses for the stateless and stateful IPv6 addresses configured in switch.
- ip-gateway	union {inet:ipv6-address;inet:ipv6-prefix;}	List of stateless gateway IPv6 addresses advertised by IP router and configured in switch to route packets to the destination IP address.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "management-ip-interface": {
            "static-ip-addresses": {
                "ip-address": [
                    "10.38.66.239"
                ]
            },
            "stateful-ip-addresses": {},
            "stateless-ipv6-addresses": {
                "ip-address": [
                    "2620:100:4:fa01:c6f5:7cff:fe02:8b08/64"
                ]
            },
            "static-ip-gateways": {
                "ip-gateway": [
                    "10.38.64.1"
                ]
            },
            "stateless-ipv6-gateways": {
                "ip-gateway": [
                    "fe80::126"
                ]
            },
            "dhcp-gateway": "",
            "static-subnet-mask": "255.255.240.0",
            "dhcp-subnet-mask": "",
            "dhcp-enabled": false,
            "ipv6-ddns-enabled": false,
            "dhcpv6-enabled": false,
            "ipv6-auto-enabled": true
        }
    }
}
```


PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-maps/backend-ports-history

Description

The historical data of backend (BE) port counters.

Parameters

Table 113: brocade-maps/backend-ports-history

Name	Type	Description
category	maps-types:port-type	A logical grouping of the BE port counters.
date	maps-types:maps-db-date -type	The date when the counters' data was cached.
crc-errors	Container	The group of ports on which the CRC (Cyclic Redundancy Check) errors were detected on BE ports. The list of ports is sorted based on the number of crc errors detected on the ports.
itw-errors	Container	The group of ports on which the ITW (Invalid Transmit Words) errors were detected on BE ports. The list of ports is sorted based on the number of ITW errors detected on the ports.
link-reset	Container	The group of ports on which the Link Resets were detected on the BE ports. The list of ports is sorted based on the number of link resets detected on the ports.
bad-os	Container	The group of ports on which the BAD OS (Invalid Ordered sets) were detected on BE ports The list of ports is sorted based on the number of invalid ordered sets detected on the ports.
long-frame	Container	The group of ports on which the long frames were detected on BE ports. The list of ports is sorted based on the number of long frame errors detected on the ports.
truncated-frame	Container	The group of ports on which the truncated frames were detected on BE ports. The list of ports is sorted based on the number of truncated frames detected on the ports.

Produces

- application/xml
- application/json

brocade-maps/credit-stall-dashboard

Description

The credit-stall congestion dashboard provides information about the monitored ports.

Note that if there is a change in switch time, then the frequency or individual state counts can be more than the defined range.

Parameters

Table 114: brocade-maps/credit-stall-dashboard

Name	Type	Description
slot-port	fibrechannel:slot-port-name-type	The slot and port information in slot/port format.
timestamp	yang:date-and-time	The system time of the credit-stall congestion data sample. The congestion data sample is available per minute and displays from latest to oldest.
state	maps-types:credit-stall-state-type	The port congestion credit-stall state of the sample for a given minute. The state can be one of the following based on the congestion experienced by port: frame-loss perf-impact medium low info no-congestion monitoring-paused
frame-loss-state-count	uint8	The number of times the port is in the frame-loss state for a given minute.
perf-impact-state-count	uint8	The number of times the port is in the perf-impact state for a given minute.
medium-state-count	uint8	The number of times the port is in the medium state for a given minute.
low-state-count	uint8	The number of times the port is in the low state for a given minute.
info-state-count	uint8	The number of times the port is in the info state for a given minute.
frequency	uint8	The number of times the port is in the congestion state for a given minute.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
```

```
  <credit-stall-dashboard>
   <slot-port>5/0</slot-port>
   <timestamp>2020-02-26T15:20:00-07:00</timestamp>
   <state>no-congestion</state>
   <frequency>0</frequency>
  </credit-stall-dashboard>
 ...
  <credit-stall-dashboard>
   <slot-port>8/63</slot-port>
   <timestamp>2020-02-26T11:55:00-07:00</timestamp>
   <state>no-congestion</state>
   <frequency>0</frequency>
  </credit-stall-dashboard>
 </Response>
```

brocade-maps/dashboard-history

Description

The historical data of various counters.

Parameters

Table 115: brocade-maps/dashboard-history

Name	Type	Description
category	maps-types:port-type	A logical grouping of the counters.
date	maps-types:maps-db-date-type	The date when the counters' data was cached.
crc-errors	Container	The group of ports on which the CRC (Cyclic Redundancy Check) errors were detected. The list of ports is sorted based on number of errors detected on the ports.
itw-errors	Container	The group of ports on which the ITW (Invalid Transmit Words) errors were detected. The list of ports is sorted based on number of errors detected on the ports.
port-loss	Container	Deprecated: Use the loss-of-sync leaf instead. The group of ports on which the loss of sync errors were seen. The list of ports is sorted based on number of errors detected on the ports.
loss-of-sync	Container	The group of ports on which the loss of sync errors were seen. The list of ports is sorted based on number of errors detected on the ports.
link-failure	Container	The group of ports on which the Link Failure errors were detected. The list of ports is sorted based on number of errors detected on the ports.
signal-loss	Container	The group of ports on which the Signal Loss errors were detected. The list of ports is sorted based on number of errors detected on the ports.

Name	Type	Description
protocol-errors	Container	The group of ports on which the Protocol Errors were detected. The list of ports is sorted based on number of errors detected on the ports.
state-change	Container	The group of ports on which state changes were detected. The list of ports is sorted based on number of state changes detected on the ports.
link-reset	Container	The group of ports on which the Link Resets were detected. The list of ports is sorted based on number of link resets detected on the ports.
c3-tx-to	Container	The group of ports on which the Class-3 frame time-outs were detected. The list of ports is sorted based on number of frames timed-out on the ports.
rx-perf	Container	A list of ports sorted based on percentage of RX bandwidth used on the port.
tx-perf	Container	A list of ports sorted based on percentage of TX bandwidth used on the port.
utilization	Container	A list of ports sorted based on percentage of total bandwidth (RX + TX) used on the port.
bottleneck-duration	Container	The bottlenecked duration in seconds.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <dashboard-history>
  <category>fc-port</category>
  <date>current</date>
  <crc-errors/>
  <itw-errors/>
  <port-loss/>
  <loss-of-sync/>
  <link-failure>
   <port-data>8/56:7</port-data>
   <port-data>7/38:1</port-data>
   <port-data>7/35:1</port-data>
  </link-failure>
  <signal-loss>
```

```
    <port-data>7/4:4</port-data>
    <port-data>7/35:1</port-data>
    <port-data>7/38:1</port-data>
   </signal-loss>
   <protocol-errors/>
   <state-change>
    <port-data>8/56:14</port-data>
    <port-data>8/57:14</port-data>
 ...
    <port-data>8/58:1</port-data>
    <port-data>8/59:1</port-data>
   </state-change>
   <link-reset>
    <port-data>8/56:17</port-data>
    <port-data>8/57:9</port-data>
 ...
    <port-data>8/59:2</port-data>
    <port-data>7/6:1</port-data>
   </link-reset>
   <c3-tx-to/>
   <rx-perf>
    <port-data>7/42:4.49</port-data>
    <port-data>8/59:2.99</port-data>
    <port-data>8/58:1.14</port-data>
   </rx-perf>
   <tx-perf>
    <port-data>7/42:4.51</port-data>
    <port-data>8/59:3.71</port-data>
    <port-data>8/58:1.13</port-data>
   </tx-perf>
   <utilization>
    <port-data>7/42:4.50</port-data>
    <port-data>8/59:3.35</port-data>
    <port-data>8/58:1.13</port-data>
   </utilization>
  </dashboard-history>
 ...
</Response>
```

brocade-maps/dashboard-misc

Description

The dashboard miscellaneous information (such as start time and operation).

Parameters

Table 116: brocade-maps/dashboard-misc

Name	Type	Description
maps-start-time	yang:date-and-time	The MAPS start time. MAPS is restartable service which means the start time could be different than system up time.
clear-data	boolean	Whether to clear the dashboard data. true : Clears the dashboard data. false: Does not clear the dashboard data. NOTE: A write-only leaf node and is not present in a GET request.
congestion-db-start-time	yang:date-and-time	The congestion dashboard start time.
fenced-ports	Container	A list of fenced ports.
- port	string	All ports that are disabled due to a FENCE action taken by MAPS rule violations.
decommissioned-ports	Container	A list of decommissioned ports.
- port	string	All ports that are decommissioned due to an action taken by MAPS rule violations.
quarantined-ports	Container	A list of quarantined ports.
- port	string	All ports that are quarantined due to an action taken by MAPS rule violations.
fenced-circuits	Container	A list of fenced circuits.
- circuit	string	All circuits that are fenced due to an action taken by MAPS rule violations.
pid-it-flows	Container	A list of the top zoned PIDs.
- pid	string	The top zoned PIDs are the five PIDs with the highest fan-in or fan-out ratio.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
  <?xml version="1.0"?>
 <Response>
  <dashboard-misc>
   <maps-start-time>Wed Jun 27 07:35:06 2018</maps-start-time>
  </dashboard-misc>
```

```
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-maps/dashboard-rule

Description

A list of dashboards. The dashboard enables you to view the events or rules that were triggered and the objects on which the rules were triggered over a specified period of time. You can also clear the dashboard data.

You can view a triggered rules list for the last 7 days. You need the rule list to get the complete picture of switch operation.
The dashboard data provides two views of the operating state: the state since midnight and the state for the last 7 days.
For both the views, you need to complete details of each rule triggered and all of the rule data.
There are several logical categories in which to classify the rules.
Each category can have maximum of five rules per hour, and each rule can have a maximum of five objects.

Parameters

Table 117: brocade-maps/dashboard-rule

Name	Type	Description
category	maps-types:maps-dashboard-category-type	The dashboard category. Each rule can belong to only one category.
name	maps-types:maps-rule-name-type	The rule name.
triggered-count	uint32	The number of times the rule was triggered for the category.
time-stamp	yang:date-and-time	The date and time that the rule was last triggered.
repetition-count	uint32	The number of times a rule was triggered. The same rule can be triggered multiple times for the same or different objects. For example, if the defALL_D_PORTSCRC_10 rule is triggered 20 times in an hour for different objects, then the repetition-count is 20.
objects	Container	The objects that violated the rule. For example port, circuit.

Name	Type	Description
- object	string	The object that violated the rule. For example, port, circuit. The object format is as follows: <element>:<value>. For example, 'F-Port 10:90' and 'U-Port 11:11'. The element can be a port with type, power supply, chassis, and so on. Each rule can have a maximum of five objects.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
 <?xml version="1.0"?>
<Response>
 <dashboard-rule>
  <category>port-health</category>
  <triggered-count>10</triggered-count>
  <name>defALL_16GSWL_SFPCURRENT_2</name>
  <time-stamp>2020-05-18T22:36:00-06:00</time-stamp>
  <repetition-count>2</repetition-count>
  <objects>
   <object>SFP 1:0</object>
   <object>SFP 1:0</object>
  </objects>
 </dashboard-rule>
...
 <dashboard-rule>
  <category>fabric-state-changes</category>
  <triggered-count>5</triggered-count>
  <name>defSWITCHFLOGI_4</name>
  <time-stamp>2020-05-18T22:31:30-06:00</time-stamp>
  <repetition-count>5</repetition-count>
  <objects>
   <object>Switch:11</object>
   <object>Switch:5</object>
   <object>Switch:10</object>
   <object>Switch:5</object>
   <object>Switch:5</object>
  </objects>
 </dashboard-rule>
</Response>
```

brocade-maps/fpi-profile

Description

The Fabric Performance Impact (FPI) profile. This container enables you to create and manage FPI monitoring thresholds. The FPI profile is a set of FPI states (such as performance impact, frame loss, and oversubscribed) and the thresholds (transmit queue latency (TXQL) and credit-zero) for each state. The FPI rules in the MAPS policy obtains the thresholds for each state from the active FPI profile.
The FPI profile is per logical switch.

Parameters

Table 118: brocade-maps/fpi-profile

Name	Type	Description
name	maps-types:maps-policy-name-type	The FPI profile name.
predefined-profile	boolean	Whether the profile is predefined or user-defined. Fabric OS ships with one predefined profile (dflt_fpi_profile). true : This profile is predefined. false: This profile is user-defined.
performance-impact	Container	The performance-impact FPI state is due to a TXQL or credit-zero stats violation.
- transmit-queue-latency	uint32	The TXQL threshold in milliseconds.
- credit-zero-percentage-1-sec	uint32	The credit-zero buffer stats threshold percentage in 1-second.
- credit-zero-percentage-5-sec	uint32	The credit-zero buffer stats threshold percentage in 5-second.
- credit-zero-percentage-10-sec	uint32	The credit-zero buffer stats threshold percentage in 10-second.
frame-loss	Container	The frame loss FPI state is due to frame transmit timeout or high TXQL that could possibly lead to frame drop condition.
- transmit-queue-latency	uint32	The TXQL threshold in milliseconds.
port-oversubscription	Container	The port-oversubscription FPI state is due to high TXQL than regular traffic condition and more than 70% TX utilization.
- transmit-percentage	uint32	The transmit (TX) utilization in percentage.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fpi-profile>
        <name>dflt_fpi_profile</name>
        <port-oversubscription>
            <transmit-percentage>70</transmit-percentage>
        </port-oversubscription>
        <performance-impact>
            <transmit-queue-latency>10</transmit-queue-latency>
            <credit-zero-percentage-1-sec>70</credit-zero-percentage-1-sec>
            <credit-zero-percentage-5-sec>50</credit-zero-percentage-5-sec>
            <credit-zero-percentage-10-sec>30</credit-zero-percentage-10-sec>
        </performance-impact>
        <frame-loss>
            <transmit-queue-latency>80</transmit-queue-latency>
        </frame-loss>
        <predefined-profile>true</predefined-profile>
    </fpi-profile>
</Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-maps/gigabit-ethernet-ports-history

Description

The historical data of GE port counters.

Parameters

Table 119: brocade-maps/gigabit-ethernet-ports-history

Name	Type	Description
category	maps-types:port-type	A logical grouping of the GE port counters.
date	maps-types:maps-db-date-type	The date when the counters' data was cached.
ge-crc-errors	Container	The group of ports on which the CRC (Cyclic Redundancy Check) errors on GE ports were detected. The list of ports is sorted based on the number of errors detected on the ports.
ge-signal-loss	Container	The group of ports on which the Signal Loss errors were detected on GE ports. The list of ports is sorted based on the number of errors detected on the ports.

Produces

- application/xml
- application/json

brocade-maps/group

Description

A list of groups to be monitored using the same set of thresholds.

For example, you can create a group of ports that behave in a similar manner, such as UNIX ports or long-distance ports. Note that all elements (ports, FCIP circuits, or SFPs) in a group must be of the same type. By creating a group of similar elements, you can manage these elements as a single entity.

You can create up to 64 user-defined groups per logical switch.

Config replay operation is supported on a predefined group only if is-modifiable leaf is true.

Parameters

Table 120: brocade-maps/group

Name	Type	Description
name	maps-types:maps-group-name-type	The group name.
group-type	maps-types:maps-group-type-type	The group type.

Name	Type	Description
group-feature	maps-types:maps-group-feature-type	The group feature. When no group feature is assigned to the group, the value is empty.
feature-pattern	string	The feature pattern.
is-predefined	boolean	Whether the group is a system-defined or user-defined group. You cannot delete or modify system-defined groups; however, you can augment the ports managed by the group. true : The group is system-defined. false: The group is user-defined.
is-modifiable	boolean	Whether you can modify the group. true : The system-defined group can be modified. false: The system-defined group cannot be modified.
members	Container	A list of members (such as, node-wwn or port-name).
- member	string	A member of the group. Valid values are: <slot>/<port> <slot>/<sfp> <slot>/<asic> <slot>/<port>/<circuit> <port> <slot or blade> <sfp> <switch> <fan index> <temperature sensor index> <power supply index> 'All Pids monitored' 'Monitored Flow'

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <group>
  <name>ALL_PORTS</name>
  <group-type>fc-port</group-type>
  <members>
   <member>5/0</member>
   <member>5/1</member>
```

```
    <member>5/2</member>
    <member>5/3</member>
    <member>5/4</member>
    <member>5/5</member>
    <member>5/6</member>
...
  <group>
   <name>ALL_FCOE_100G_SR4_QSFP</name>
   <group-type>sfp</group-type>
   <members>
    <member/>
   </members>
   <is-predefined>true</is-predefined>
   <is-augmentable>false</is-augmentable>
  </group>
  <group>
   <name>ALL_OTHER_F_PORTS_1</name>
   <group-type>fc-port</group-type>
   <members>
    <member>8/5</member>
   </members>
   <is-predefined>false</is-predefined>
   <is-augmentable>false</is-augmentable>
  </group>
</Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-maps/maps-config

Description

The MAPS configuration for the switch. You can perform the following

MAPS configurations using this container:

1. View the current MAPS configuration.

2. Define the actions to take on the switch when a threshold is

triggered.

3. Specify the e-mail addresses to which the alerts are sent.

4. Delete all user-defined MAPS configurations related to rules,

groups, policies, and so on.

Parameters

Table 121: brocade-maps/maps-config

Name	Type	Description
decommission-cfg	enumeration	The decommission behavior: with-disable or impair.
sender-address	string	The e-mail address of the sender.
domain-name	string	The domain name. Enter the name as none to clear it.
relay-ip-address	union {inet:ip-address;fibrechannel:brocade-empty-string-type;inet:domain-name;}	The relay IP address. Enter the address as none to clear it.
quiet-time	uint32	Configuration of global quiet time will be deprecated from v9.2.0. The global quiet time. Rules in the policy are not triggered until quiet time has expired.
f-port-fpi-profile	maps-types:maps-policy-name-type	The F_Port FPI active profile name.
e-port-fpi-profile	maps-types:maps-policy-name-type	The E_Port FPI active profile name.
notification	maps-types:notification-type	The MAPS Notification setting.
actions	Container	The global MAPS actions list.
- action	maps-types:maps-generic-action-type	The MAPS actions. You can enable one or more actions globally at the switch level or per rule.
recipient-address-list	Container	The recipient list for e-mail alerts.
- recipient-address	string	The e-mail address (such as john@mmm.com or psingh@xyz.com) for recipients of e-mail alerts.
test-email	Container	The test email. Note: write-only container and will not be part of GET request.
- subject	string	The subject for the email alert. Note: write-only leaf and will not be part of GET request.
- body	string	The message for the email alert. Note: write-only leaf and will not be part of GET request.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "maps-config": {
            "recipient-address-list": {},
            "relay-ip-address": "",
            "domain-name": "",
            "actions": {
                "action": [
                    "sfp-marginal",
                    "sw-critical",
                    "sw-marginal",
                    "fpin"
                ]
            },
            "sender-address": "",
            "decommission-cfg": "impair",
            "quiet-time": 0,
            "f-port-fpi-profile": "dflt_fpi_profile",
            "e-port-fpi-profile": "dflt_fpi_profile",
            "notification": "default"
        }
    }
}
```


PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-maps/maps-device-login

Description

The ports connected to the Fabric Interconnect (FI).

Parameters

Table 122: brocade-maps/maps-device-login

Name	Type	Description
node-wwn	fibrechannel:wwn-type	The node WWN of the Fabric Interconnect (FI).
state	maps-types:port-group-state-type	The current state of the port group.
master-port	string	The master port. The master port used in the alert.

Name	Type	Description
ports	Container	A port group that contains zero or more ports. These are ports connected to the Fabric Interconnect(FI).
- port	string	A member port.

Produces

- application/xml
- application/json

brocade-maps/maps-policy

Description

The MAPS policy. This container enables you to create and manage monitoring policies. A MAPS policy is a set of rules that define thresholds for measures and actions to take when a threshold is triggered. When you enable a policy, all of the rules in the policy are in effect. A switch can have multiple policies.

Parameters

Table 123: brocade-maps/maps-policy

Name	Type	Description
name	maps-types:maps-policy-name-type	The MAPS policy name.
is-active-policy	boolean	Whether the policy is active. You can configure multiple policies; however, only one policy can be active at a time. true : This policy is active. false: This policy is not active. If sent in a configuration operation, this is treated as a 'no-op' and the switch does not take any action as MAPS does not support disabling a policy.
is-predefined-policy	boolean	Whether the policy is predefined or user-defined. Fabric OS ships with 4 predefined policies (dflt_conservative_policy, dflt_aggressive_policy, dflt_moderate_policy, and dflt_base_policy). true : This policy is predefined. false: This policy is user-defined.
rule-list	Container	A list of rules in the policy.
- rule	maps-types:maps-rule-name-type	The rule name.
system-rules	Container	A list of system policy rules. The system policy rule list is added to current active policy. Note that the system policy rule is omitted from this list if the current active policy has a corresponding custom rule or default system rule added directly to custom policy.

Name	Type	Description
- rule	maps-types:maps-rule-name-type	The rule name.
unmonitored-rules	Container	A list of unmonitored rules in active policy rule list. The active policy rule(s) are added to the unmonitored rule list for any validation failed during policy activation.
- rule	maps-types:maps-rule-name-type	The rule name.
deleted-rules	Container	A list of default rules deleted from default policy. From FOS920, can delete the rules from default policy. Deleted rules, though present in the policy, but not monitored and will be be empty for user policy.
- rule	maps-types:maps-rule-name-type	The rule name.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "maps-policy": [
            {
                "name": "dflt_aggressive_policy",
                "rule-list": {
                    "rule": [
                        "defALL_10GLWL_SFPCURRENT_8",
                        "defALL_10GLWL_SFPCURRENT_95",
                        ...
                        "def_IO_ERROR",
                        "def_RESERVE"
                    ]
                },
                "system-rules": {},
                "unmonitored-rules": {},
                "deleted-rules": {},
                "is-predefined-policy": true,
                "is-active-policy": false
            },
            {
```

```
        "name": "dflt_moderate_policy",
        "rule-list": {
            "rule": [
                "defALL_10GLWL_SFPCURRENT_8",
                "defALL_10GLWL_SFPCURRENT_95",
                ...
                "def_IO_ERROR",
                "def_RESERVE"
            ]
        },
        "system-rules": {},
        "unmonitored-rules": {},
        "deleted-rules": {},
        "is-predefined-policy": true,
        "is-active-policy": false
    },
    {
        "name": "dflt_conservative_policy",
        "rule-list": {
            "rule": [
                "defALL_10GLWL_SFPCURRENT_8",
                "defALL_10GLWL_SFPCURRENT_95",
                ...
                "def_IO_ERROR",
                "def_RESERVE"
            ]
        },
        "system-rules": {},
        "unmonitored-rules": {},
        "deleted-rules": {},
        "is-predefined-policy": true,
        "is-active-policy": false
    },
    {
        "name": "dflt_base_policy",
        "rule-list": {
            "rule": [
                "defALL_E_PORTSROUTING_ERR_1",
                "defALL_E_PORTSROUTING_ERR_ROR_2",
                ...
                "defCHASSISSYSTEM_TEMP_MARG",
                "defSWITCHASC_UPLOAD_FAILURE_3"
            ]
        },
        "system-rules": {
            "rule": [
                "defCHASSISTRUFOS_CERT_EXPIRED",
                "defCHASSISTRUFOS_CERT_DAYS_TO_EXPIRE_60",
                ...
                "defCHASSISMEMORY_USAGE_STATE_IN_RANGE",
                "defCHASSISMEMORY_USAGE_STATE_WARN"
            ]
        },
```

```
                "unmonitored-rules": {},
                "deleted-rules": {},
                "is-predefined-policy": true,
                "is-active-policy": true
            },
            {
                "name": "dflt_always_active_policy",
                "rule-list": {
                    "rule": [
                        "defCHASSISMEMORY_USAGE_STATE_CRIT",
                        "defCHASSISMEMORY_USAGE_STATE_IN_RANGE",
                        ...
                        "defCHASSISTRUFOS_CERT_EXPIRED",
                        "defCHASSISTRUFOS_CERT_INSTALLED"
                    ]
                },
                "system-rules": {},
                "unmonitored-rules": {},
                "deleted-rules": {},
                "is-predefined-policy": true,
                "is-active-policy": true
            }
        ]
    }
}
```


DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-maps/monitoring-system-matrix

Description

A list of monitoring systems. Each monitoring system can support different time bases, actions, and thresholds. Some monitoring systems are supported only on specific systems. For example, circuit or tunnel monitoring systems are only supported on extension platforms.

Parameters

Table 124: brocade-maps/monitoring-system-matrix

Name	Type	Description
monitoring-system	maps-types:maps-monitoring-system-type	The monitoring system name.
group-type	maps-types:maps-group-type-type	The group type.
dashboard-category	maps-types:maps-dashboard-category-type	The dashboard category of the monitoring system.
is-read-only	boolean	Whether the monitoring system is read only.
monitored-logical-switch	enumeration	Whether the monitoring system is in all logical switches or only the default logical switch.
is-rule-on-rule-supported	boolean	Whether rule on rule is supported. true : Rule on rule is supported. false: Rule on rule is not supported.
is-quiet-time-supported	boolean	Whether quiet time is supported. true : Quiet time is supported false : Quiet time is not supported
minimum-quiet-time	uint32	The minimum quiet time.
monitoring-type	maps-types:monitoring-type-type	The monitoring type (event based or poll based) for the monitoring system.
data-type	maps-types:data-type-type	The data type support. Each monitoring system supports different data types for thresholds.
description	string	A description of the monitoring system.
unit	maps-types:maps-data-unit	The data unit.
data-range	string	The data range.
adaptive-notification-supported	boolean	Whether Adaptive Notification feature is supported. When this is set, QT is not supported for the monitoring system. true : Adaptive Notification is supported false : Adaptive Notification is not supported
base-time-bases	Container	The time bases.
- time-base	maps-types:maps-time-base-type	A list of the supported time bases for the monitoring system.
rule-on-rule-time-bases	Container	The supported rule on rules time bases for the monitoring system.
- rule-on-rule-time-base	maps-types:maps-time-base-type	A list of the supported time bases for rule on rules for the monitoring system.
actions	Container	The global MAPS actions list.
- action	maps-types:maps-generic-action-type	A list of the MAPS actions defined for this monitoring system.
logical-operators	Container	The supported operations.
- logical-operator	maps-types:maps-logical-operator-type	The logical operator.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "monitoring-system-matrix": [
            {
                "monitoring-system": "CRC",
                "group-type": "fc-port",
                "data-type": "unsigned-int32",
                "data-range": "0-999999999",
                "dashboard-category": "port-health",
                "description": "CRC errors (CRC)",
                "unit": "CRCs",
                "base-time-bases": {
                    "time-base": [
                        "minute",
                        "hour",
                        "day"
                    ]
                },
                "actions": {
                    "action": [
                        "raslog",
                        "snmp-trap",
                        "e-mail",
                        "port-fence",
                        "decommission",
                        "fms",
                        "fpin"
                    ]
                },
                "is-quiet-time-supported": true,
                "is-rule-on-rule-supported": true,
                "rule-on-rule-time-bases": {
                    "rule-on-rule-time-base": [
                        "hour",
                        "day",
                        "week"
                    ]
                },
                "monitoring-type": "poll-based",
```

```
        "monitored-logical-switch": "all-logical-switches",
        "logical-operators": {
            "logical-operator": [
                "g",
                "ge"
            ]
        },
        "minimum-quiet-time": 60,
        "is-read-only": false,
        "adaptive-notification-supported": true
    },
    ...
    {
        "monitoring-system": "STATE_CHG",
        "group-type": "circuit-qos",
        "data-type": "unsigned-int32",
        "data-range": "0-999999999",
        "dashboard-category": "extension-health",
        "description": "Circuit QOS State Change (STATE_CHG)",
        "unit": "not-applicable",
        "base-time-bases": {
            "time-base": [
                "minute",
                "hour",
                "day"
            ]
        },
        "actions": {
            "action": [
                "raslog",
                "snmp-trap",
                "e-mail",
                "port-fence",
                "fms"
            ]
        },
        "is-quiet-time-supported": true,
        "is-rule-on-rule-supported": true,
        "rule-on-rule-time-bases": {
            "rule-on-rule-time-base": [
                "hour",
                "day",
                "week"
            ]
        },
        "monitoring-type": "poll-based",
        "monitored-logical-switch": "all-logical-switches",
        "logical-operators": {
            "logical-operator": [
                "g",
                "ge"
            ]
        },
```

```
                "minimum-quiet-time": 60,
                "is-read-only": false,
                "adaptive-notification-supported": true
            }
        ]
    }
}
```


brocade-maps/oversubscription-dashboard

Description

The oversubscription dashboard provides information about the monitored ports.

Note that if there is a change in switch time, then the frequency count can be more than the defined range.

Parameters

Table 125: brocade-maps/oversubscription-dashboard

Name	Type	Description
slot-port	fibrechannel:slot-port-name-type	The slot and port information in slot/port format.
timestamp	yang:date-and-time	The system time of the congestion data sample. The congestion data sample is available per minute and displays from latest to oldest.
state	maps-types:oversubscription-state-type	The port oversubscription state of the sample for a given minute. The state can be one of the following based on the oversubscription experienced by port: oversubscription no-oversubscription monitoring-paused
frequency	uint8	The number of times the port is in the oversubscription state for a given minute.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <oversubscription-dashboard>
  <slot-port>5/0</slot-port>
  <timestamp>2020-02-26T15:24:00-07:00</timestamp>
  <state>no-oversubscription</state>
 </oversubscription-dashboard>
 <oversubscription-dashboard>
  <slot-port>5/0</slot-port>
  <timestamp>2020-02-26T15:23:00-07:00</timestamp>
  <state>no-oversubscription</state>
 </oversubscription-dashboard>
...
 <oversubscription-dashboard>
  <slot-port>8/63</slot-port>
  <timestamp>2020-02-26T11:56:00-07:00</timestamp>
  <state>no-oversubscription</state>
 </oversubscription-dashboard>
 <oversubscription-dashboard>
  <slot-port>8/63</slot-port>
  <timestamp>2020-02-26T11:55:00-07:00</timestamp>
  <state>no-oversubscription</state>
 </oversubscription-dashboard>
</Response>
```

brocade-maps/paused-cfg

Description

A list of elements or element groups on which you want to pause or restart monitoring. You can only pause or restart monitoring of FC ports, FCIP circuits (and QOS), SFPs, and switch groups. Use the keyword 'all' to pause or restart monitoring for all supported groups.

Parameters

Table 126: brocade-maps/paused-cfg

Name	Type	Description
group-type	enumeration	The element group on which you want to pause or restart monitoring.
members	Container	A list of elements (FC ports, FCIP circuits, SFPs, switch) or all members in the group type. There must be at least one member in the list. Use the keyword 'all' to include all group types.

Name	Type	Description
- member	string	A member of the group. Valid values are: <slot>/<port> <slot>/<port>/<circuit> <slot>/<sfp> <port> <sfp>

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <paused-cfg>
  <group-type>fc-port</group-type>
  <members>
   <member>8/9</member>
   <member>8/15</member>
   <member>8/16</member>
   <member>8/17</member>
   <member>8/18</member>
   <member>6/23</member>
  </members>
 </paused-cfg>
</Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-maps/quarantined-devices

Description

Displays information about all online quarantined devices

across fabric.

Parameters

Table 127: brocade-maps/quarantined-devices

Name	Type	Description
port-index	fibrechannel:port-number-type	Port number of the switch.
port-id	fibrechannel:fcid-hex-string-type	Port identifier.
port-wwn	fibrechannel:wwn-type	Port world wide name.
locality	enumeration	Identifies whether the quarantined device is a local or a remote switch.
port-name	string	A user-friendly name assigned to a port.
zone-aliases	Container	A list of zone aliases when AG mode is not enabled.
- alias-name	zone:zoning-name-type	A zone alias name.

Produces

- application/xml
- application/json

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
        <?xml version=""1.0""?>
 <Response>
     <quarantined-devices>
         <port-index>9</port-index>
         <port-id>67840</port-id>
         <port-wwn>21:00:00:24:ff:7d:3d:b8</port-wwn>
         <locality>LOCAL</locality>
         <port-name>port9</port-name>
         <zone-aliases>
             <alias-name>bro_alias</alias-name>
             <alias-name>brocade_alias</alias-name>
             <alias-name>my_alias</alias-name>
         </zone-aliases>
     </quarantined-devices>
     ...
     <quarantined-devices>
         <port-index>497</port-index>
         <port-id>651584</port-id>
         <port-wwn>20:00:00:11:0d:70:00:00</port-wwn>
         <locality>REMOTE</locality>
         <zone-aliases>
             <alias-name>bro_alias</alias-name>
```

```
            <alias-name>brocade_alias</alias-name>
            <alias-name>my_alias</alias-name>
        </zone-aliases>
    </quarantined-devices>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-maps/rule

Description

A list of rules. You can use the rules container to configure and manage MAPS monitoring rules and to display configured rules. A rule associates a condition (threshold) with actions that are triggered when the specified condition is reached. A rule must be in the enabled MAPS policy to be active. When you modify a rule in the enabled MAPS policy, the rule does not take effect until you re-enable the policy.

Parameters

Table 128: brocade-maps/rule

Name	Type	Description
name	maps-types:maps-rule-name-type	The rule name.
is-rule-on-rule	maps-types:maps-rule-type	The rule name. A rule can be one of two types: base or rule-on-rule. A base rule monitors the statistics or FRUs whereas a rule-on-rule monitors the base rule.
monitoring-system	maps-types:maps-monitoring-system-type	The statistic or error to be monitored (CRC, ITW, PS_STATE, and so on).
time-base	maps-types:maps-time-base-type	The time interval between two samples to be compared.
logical-operator	maps-types:maps-logical-operator-type	The relational operation to be used in evaluating the condition.
threshold-value	maps-types:threshold-value-type	The threshold value. Thresholds are the values at which potential problems might occur. For example, in configuring a port threshold, you can select a specific value at which an action is triggered because of too many threshold violations.
group-name	maps-types:maps-group-name-type	The group name (such as ALL_PORTS).

Name	Type	Description
is-predefined	boolean	Whether the group is a system-defined or user-defined group. You cannot delete system-defined groups; however, you can augment the ports managed by the group. true : The group is system-defined. false: The group is user-defined.
event-severity	maps-types:maps-event-severity-type	The user-configured severity. If the value is empty then system assigned the severity to the rules. For more information about the system-defined severity, refer to the MAPS User Guide.
toggle-time	uint32	The port's toggle time.
quiet-time	uint32	Configuration of rule quiet time will be deprecated from v9.2.0. The quiet time. The rule is not triggered until quiet time has expired. This leaf node is not supported for IT_FLOW monitoring system.
quiet-time-clear	boolean	Whether to clear quiet time. true : Clears the quiet time from the rule. false: Does not clear the quiet time from the rule. Use a quiet-time leaf with a 0 value to clear the quiet time. NOTE: A write-only leaf node and is not present in a GET request.
un-quarantine-timeout	uint32	The un-quarantine timeout.
un-quarantine-clear	boolean	Whether to clear the un-quarantine timeout. true : Clears the quiet time from the rule. false: Does not clear the quiet time from the rule. NOTE: A write-only leaf node and is not present in a GET request.
actions	Container	The MAPS actions.
- action	maps-types:maps-generic-action-type	A MAPS action.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <rule>
  <name>defALL_100G_QSFPCURRENT_10</name>
  <is-rule-on-rule>false</is-rule-on-rule>
  <monitoring-system>CURRENT</monitoring-system>
  <time-base>NONE</time-base>
```

```
   <logical-operator>ge</logical-operator>
   <threshold-value>10</threshold-value>
   <group-name>ALL_FCOE_100G_SR4_QSFP</group-name>
   <actions>
    <action>raslog</action>
    <action>snmp-trap</action>
    <action>e-mail</action>
    <action>sfp-marginal</action>
   </actions>
   <is-predefined>true</is-predefined>
  </rule>
 ...
  <rule>
   <name>ALL_PORTS_IO_FRAME_LOSS_UNQUARtest</name>
   <is-rule-on-rule>false</is-rule-on-rule>
   <monitoring-system>DEV_LATENCY_IMPACT</monitoring-system>
   <time-base>NONE</time-base>
   <logical-operator>eq</logical-operator>
   <threshold-value>IO_FRAME_LOSS</threshold-value>
   <group-name>ALL_PORTS</group-name>
   <actions>
    <action>raslog</action>
    <action>snmp-trap</action>
    <action>e-mail</action>
    <action>port-toggle</action>
    <action>un-quarantine</action>
    <action>sddq</action>
   </actions>
   <is-predefined>false</is-predefined>
   <toggle-time>2</toggle-time>
   <quiet-time>60</quiet-time>
   <un-quarantine-timeout>1</un-quarantine-timeout>
  </rule>
 </Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-maps/switch-status-policy-report

Description

The Switch Status Policy (SSP) report. The SSP report provides the

overall health status of the switch and includes enough information for you to investigate further, if necessary.

Parameters

Table 129: brocade-maps/switch-status-policy-report

Name	Type	Description
switch-health	maps-types:ssp-state-type	The overall status of the switch. The switch state is determined by the state of one or more of its components.
power-supply-health	maps-types:ssp-state-type	The state of the power supplies.
fan-health	maps-types:ssp-state-type	The state of the fans.
wwn-health	maps-types:ssp-state-type	The state of the WWN cards.
temperature-sensor-health	maps-types:ssp-state-type	The state of the temperature sensors.
ha-health	maps-types:ssp-state-type	The state of the high availability state (both control processors are in sync).
control-processor-health	maps-types:ssp-state-type	The state of the control processors.
core-blade-health	maps-types:ssp-state-type	The state of the core blades.
blade-health	maps-types:ssp-state-type	The state of the application blades.
flash-health	maps-types:ssp-state-type	The state of the flash usage. Flash usage could be above the threshold or limit.
marginal-port-health	maps-types:ssp-state-type	The state of the marginal ports.
faulty-port-health	maps-types:ssp-state-type	The state of the faulty ports.
missing-sfp-health	maps-types:ssp-state-type	The state of the missing SFPs.
error-port-health	maps-types:ssp-state-type	The state of the error ports.
expired-certificate-health	maps-types:ssp-state-type	The state of the expired certificate.
airflow-mismatch-health	maps-types:ssp-state-type	The state of the air flow. This parameter monitors the air flow direction of the power supply fan FRUs and blower FRUs and generates an alert if there is a mismatch in the air flow direction of any two power supply fans or any two blowers.
marginal-sfp-health	maps-types:ssp-state-type	The state of SFPs. If a certain percentage of SFPs are in SFP_MARGINAL, the switch state is moved to marginal or down.
trusted-fos-certificate-health	maps-types:ssp-state-type	The Trusted FOS Certificate state, the switch state is moved to marginal or down when the certificate is expired. Obsolete because MAPS no longer considers trusted-fos-certificate-health as a Switch Status Policy Monitoring.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <switch-status-policy-report>
  <switch-health>healthy</switch-health>
  <power-supply-health>healthy</power-supply-health>
  <fan-health>healthy</fan-health>
  <wwn-health>healthy</wwn-health>
  <temperature-sensor-health>healthy</temperature-sensor-health>
  <ha-health>healthy</ha-health>
  <control-processor-health>healthy</control-processor-health>
  <core-blade-health>healthy</core-blade-health>
  <blade-health>healthy</blade-health>
  <flash-health>healthy</flash-health>
  <marginal-port-health>healthy</marginal-port-health>
  <faulty-port-health>healthy</faulty-port-health>
  <missing-sfp-health>healthy</missing-sfp-health>
  <error-port-health>healthy</error-port-health>
  <expired-certificate-health>healthy</expired-certificate-health>
  <airflow-mismatch-health>healthy</airflow-mismatch-health>
  <marginal-sfp-health>healthy</marginal-sfp-health>
  <trusted-fos-cert-health>healthy</trusted-fos-cert-health>
 </switch-status-policy-report>
</Response>
```

brocade-maps/system-resources

Description

The system resources (such as CPU, RAM, and flash memory usage).

Note that usage is not real time and may be delayed up to 2 minutes.

Parameters

Table 130: brocade-maps/system-resources

Name	Type	Description
cpu-usage	uint32	The percentage of CPU usage.
memory-usage	uint32	The percentage of memory usage.
total-memory	uint32	The total memory usage.
flash-usage	uint32	The percentage of flash usage.
free-kernel-memory	uint32	The percentage of available kernel memory..

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "system-resources": {
            "cpu-usage": 7,
            "memory-usage": 15,
            "total-memory": 8150348,
            "flash-usage": 24,
            "free-kernel-memory": 0
        }
    }
}
```

brocade-media/media-rdp

Description

A list of switch port and peer port details.

Displays Small Form-factor Pluggable (SFP) transceiver information.

These SFP parameters provide extended information that describes the

SFP capabilities, interfaces, manufacturer, and so on.

Parameters

Table 131: brocade-media/media-rdp

Name	Type	Description
part-number	fru:part-number-type	The part number assigned by the organization responsible for producing or manufacturing the media.
serial-number	fru:serial-number-type	The factory's serial number for the blade.
vendor-name	vendor-name-type	The vendor name of the media.
vendor-revision	vendor-revision-type	The vendor revision number of the media.
date-code	date-code-type	The vendor's manufacturing date code.
name	interface-type	The name of the SFP interface in the format interface-id/slot/port. The name of the interface.

Name	Type	Description
identifier	string	The type of serial transceiver inserted in the port module.
connector	string	The connector type of the media.
nominal-baud-rate	int32	The nominal signalling rate of the media.
baud-rate-maximum	int32	The maximum baud rate.
baud-rate-minimum	int32	The minimum baud rate.
transmission-type	transmission-types	The media transmission type.
fiber-9um-link-length	uint32	The link length supported for 9um fiber.
fiber-9m-link-length	uint32	The link length supported for 9m fiber.
multimode-fiber-om2-50um-length	uint32	The link length supported for 50um OM2 multimode fiber.
multimode-fiber-om3-50um-length	uint32	The link length supported for 50um OM3 multimode fiber.
multimode-fiber-om4-50um-length	uint32	The link length supported for 50um OM4 multimode fiber.
multimode-fiber-625um-length	uint32	The link length supported for 625um OM1 multimode fiber.
copper-link-length	uint32	The minimum link length supported by the transceiver with applicable standards using copper cable.
enhanced-options	fibrechannel:brocade-hex-string-type	The optional digital diagnostic enhanced features implemented in the transceiver.
digital-diagnostics-type	fibrechannel:brocade-hex-string-type	The type of diagnostic monitoring implemented in the media.
media-optional-signal-value	string	The media optional signal value.
status-control	fibrechannel:brocade-hex-string-type	The current status of the feature implemented by the media.
electric-loopback-enabled	boolean	Whether electric loopback is enabled or disabled. true : Electric loopback is enabled. false: Electric loopback is disabled.
optical-loopback-enabled	boolean	Whether optical loopback is enabled or disabled. true : Optical loopback is enabled. false: Optical loopback is disabled.
encoding	string	The serial encoding algorithm used in transmitting data on Fibre Channel and Gigabit Ethernet. Encoding supports continuous transmission with a balanced number of ones and zeros in the code stream, and detects single bit transmission errors.
vendor-oui	string	The vendor IEEE company ID for the media. An organizationally unique identifier (OUI) is a 24-bit number that uniquely identifies a vendor, manufacturer, or other organization.
temperature	temperature-type	The temperature of the media.
rx-power	media-power-type	The receive power of the media.
tx-power	media-power-type	The transmit power of the media.
current	current-type	The transceiver current of the media.

Name	Type	Description
voltage	voltage-type	The transceiver voltage of the media.
wavelength	uint32	The laser wavelength of the media connector.
power-on-time	int32	The current value of the power-on time. This displays for 16-Gbps, 32-Gbps, or 64-Gbps SFPs and QSFPs only.
state-transitions	uint32	The number of times the physical media was removed or inserted.
last-poll-time	union {yang:date-and-time;fibrechannel:brocade-empty-string-type;}	The last poll timestamp of the media. An empty string displays when polling has not started.
peer-data-available	boolean	Whether peer data is available or not. false: Peer data not available. true: Peer data available.
remote-identifier	string	The identifier for the remote media.
remote-laser-type	string	The laser type of the remote media connector.
remote-optical-type	optical-type	The optical type of the remote media connector.
remote-media-temperature	temperature-type	The temperature of the remote media.
remote-media-rx-power	media-power-type	The receive power of the remote media.
remote-media-tx-power	media-power-type	The transmit power of the remote media.
remote-media-current	current-type	The transceiver current of the remote media.
remote-media-voltage	voltage-type	The transceiver voltage of the remote media.
optical-cable-types	Container	A list of optical cable types.
- cable-type	cable-descriptor-type	The type of optical media cable.
media-optional-signals	Container	A list of the media optional signals.
- signal	signal-descriptor-type	The optional transceiver signals implemented by the media.
media-speed-capability	Container	A list of capable speeds supported by media.
- speed	capable-speed-type	The supported transmission speeds of the media. Deprecated: Use the media-speed-capability-v2.
media-speed-capability-v2	Container	A list of capable speeds supported by media.
- speed	fibrechannel:protocol-speed-type	The supported transmission speeds of the media.
media-distance	Container	A list of distances supported by media.
- distance	string	The distance supported by the media transceiver.
alarm-flags	Container	A list of alarm flags associated with the transceiver media.
- high-alarm	fibrechannel:brocade-hex-string-type	The high alarm for the media.
- low-alarm	fibrechannel:brocade-hex-string-type	The low alarm for the media.
warning-flags	Container	A list of warning flags associated with the transceiver media.
- high-warning	fibrechannel:brocade-hex-string-type	The high warning level for the media.

Name	Type	Description
- low-warning	fibrechannel:brocade-hex-string-type	The low warning level for the media.
remote-media-speed-capability	Container	A list of capable speeds supported by media.
- speed	capable-speed-type	The supported transmission speeds of the remote port. Deprecated: Use the remote-media-speed-capability-v2.
remote-media-speed-capability-v2	Container	A list of capable speeds supported by media.
- speed	fibrechannel:protocol-speed-type	The supported transmission speeds of the remote port.
remote-optical-product-data	Container	The optical product data of the remote media.
part-number	fru:part-number-type	The part number assigned by the organization responsible for producing or manufacturing the media.
serial-number	fru:serial-number-type	The factory's serial number for the blade.
vendor-name	vendor-name-type	The vendor name of the media.
vendor-revision	vendor-revision-type	The vendor revision number of the media.
date-code	date-code-type	The vendor's manufacturing date code.
remote-media-voltage-alert	Container	A list of warning and alarm levels for remote media voltage. obsolete: Use the 'remote-media-voltage-alarm-type' container instead.
high-alarm	alert-type	The high alarm for the media.
low-alarm	alert-type	The low alarm for the media.
high-warning	alert-type	The high warning level for the media.
low-warning	alert-type	The low warning level for the media.
remote-media-signal-loss	Container	The data of upstream and downstream signal loss for remote media.
- up-stream	stream-type	The upstream signal loss data for the media.
- down-stream	stream-type	The downstream signal loss data for the media.
remote-media-temperature-alert	Container	A list of warning and alarm levels for remote media temperature. obsolete: Use the 'remote-media-temperature-alarm-type' container instead.
high-alarm	alert-type	The high alarm for the media.
low-alarm	alert-type	The low alarm for the media.
high-warning	alert-type	The high warning level for the media.
low-warning	alert-type	The low warning level for the media.
remote-media-tx-bias-alert	Container	A list of warning and alarm levels for remote media transceiver. obsolete: Use the 'remote-media-tx-bias-alarm-type' container instead.
high-alarm	alert-type	The high alarm for the media.
low-alarm	alert-type	The low alarm for the media.
high-warning	alert-type	The high warning level for the media.
low-warning	alert-type	The low warning level for the media.

Name	Type	Description
remote-media-tx-power-alert	Container	A list of warning and alarm levels for remote media transceiver power. obsolete: Use the 'remote-media-tx-power-alarm-type' container instead.
high-alarm	alert-type	The high alarm for the media.
low-alarm	alert-type	The low alarm for the media.
high-warning	alert-type	The high warning level for the media.
low-warning	alert-type	The low warning level for the media.
remote-media-rx-power-alert	Container	A list of warning and alarm levels for remote media receiver power. obsolete: Use the 'remote-media-rx-power-alarm-type' container instead.
high-alarm	alert-type	The high alarm for the media.
low-alarm	alert-type	The low alarm for the media.
high-warning	alert-type	The high warning level for the media.
low-warning	alert-type	The low warning level for the media.
remote-media-voltage-alarm-type	Container	A list of warning and alarm levels for remote media voltage.
- high-alarm	alarm-type	The high alarm for the media.
- low-alarm	alarm-type	The low alarm for the media.
- high-warning	alarm-type	The high warning level for the media.
- low-warning	alarm-type	The low warning level for the media.
remote-media-temperature-alarm-type	Container	A list of warning and alarm levels for remote media temperature.
- high-alarm	temp-alarm-type	The high alarm for the media.
- low-alarm	temp-alarm-type	The low alarm for the media.
- high-warning	temp-alarm-type	The high warning level for the media.
- low-warning	temp-alarm-type	The low warning level for the media.
remote-media-tx-bias-alarm-type	Container	A list of warning and alarm levels for remote media transceiver.
- high-alarm	alarm-type	The high alarm for the media.
- low-alarm	alarm-type	The low alarm for the media.
- high-warning	alarm-type	The high warning level for the media.
- low-warning	alarm-type	The low warning level for the media.
remote-media-tx-power-alarm-type	Container	A list of warning and alarm levels for remote media transceiver power.
- high-alarm	alarm-type	The high alarm for the media.
- low-alarm	alarm-type	The low alarm for the media.
- high-warning	alarm-type	The high warning level for the media.
- low-warning	alarm-type	The low warning level for the media.
remote-media-rx-power-alarm-type	Container	A list of warning and alarm levels for remote media receiver power.

Name	Type	Description
- high-alarm	alarm-type	The high alarm for the media.
- low-alarm	alarm-type	The low alarm for the media.
- high-warning	alarm-type	The high warning level for the media.
- low-warning	alarm-type	The low warning level for the media.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.


```
{
    "Response": {
        "media-rdp": [
            {
                "name": "fc/0/4",
                "identifier": "SFP",
                "connector": "LC",
                "transmission-type": "shortwave",
                "encoding": "PAM4",
                "vendor-name": "BROCADE",
                "vendor-oui": "00:05:1e",
                "part-number": "57-1000495-01",
                "vendor-revision": "B",
                "serial-number": "MAA12033C001895S",
                "date-code": "200812",
                "temperature": 41,
                "rx-power": 809.60,
                "tx-power": 914.50,
                "current": 6.98,
                "voltage": 3231.10,
                "wavelength": 850,
                "power-on-time": 21533,
                "media-distance": {
                    "distance": [
                        "intermediate",
                        "short"
                    ]
                },
                "optical-cable-types": {
                    "cable-type": [
```

```
            "M5"
        ]
    },
    "media-optional-signal-value": "0a3a",
    "media-speed-capability": {
        "speed": [
            16,
            32,
            64
        ]
    },
    "media-speed-capability-v2": {
        "speed": [
            "16-gfc",
            "32-gfc",
            "64-gfc"
        ]
    },
    "nominal-baud-rate": 255,
    "baud-rate-maximum": 231,
    "baud-rate-minimum": 0,
    "fiber-9um-link-length": 0,
    "fiber-9m-link-length": 0,
    "multimode-fiber-om2-50um-length": 2,
    "multimode-fiber-om3-50um-length": 7,
    "multimode-fiber-om4-50um-length": 10,
    "multimode-fiber-625um-length": 0,
    "copper-link-length": 0,
    "enhanced-options": "0xfa",
    "digital-diagnostics-type": "0x68",
    "media-optional-signals": {
        "signal": [
            "loss-of-signal",
            "tx-fault",
            "tx-disable"
        ]
    },
    "status-control": "0x30",
    "electric-loopback-enabled": false,
    "optical-loopback-enabled": false,
    "alarm-flags": {
        "high-alarm": "0x0",
        "low-alarm": "0x0"
    },
    "warning-flags": {
        "high-warning": "0x0",
        "low-warning": "0x0"
    },
    "state-transitions": 1,
    "last-poll-time": "2023-03-01T14:49:53-07:00",
    "peer-data-available": true,
    "remote-identifier": "SFP+",
    "remote-laser-type": "short wave laser",
```

```
"remote-optical-type": "optical-port-type",
"remote-media-signal-loss": {
    "up-stream": 140.90,
    "down-stream": 102.40
},
"remote-media-speed-capability": {
    "speed": [
        8,
        16,
        32
    ]
},
"remote-media-speed-capability-v2": {
    "speed": [
        "8-gfc",
        "16-gfc",
        "32-gfc"
    ]
},
"remote-optical-product-data": {
    "part-number": "AFBR-57G5MZ-ELX",
    "serial-number": "AD1921G020T",
    "vendor-name": "AVAGO",
    "vendor-revision": "",
    "date-code": "190524"
},
"remote-media-temperature": 42,
"remote-media-rx-power": 773.60,
"remote-media-tx-power": 912.00,
"remote-media-current": 7.50,
"remote-media-voltage": 3338.50,
"remote-media-voltage-alarm-type": {
    "high-alarm": 3630,
    "low-alarm": 2970,
    "high-warning": 3465,
    "low-warning": 3100
},
"remote-media-temperature-alarm-type": {
    "high-alarm": 75,
    "low-alarm": -5,
    "high-warning": 70,
    "low-warning": 0
},
"remote-media-tx-bias-alarm-type": {
    "high-alarm": 10000,
    "low-alarm": 2000,
    "high-warning": 8500,
    "low-warning": 3000
},
"remote-media-tx-power-alarm-type": {
    "high-alarm": 3467,
    "low-alarm": 48,
    "high-warning": 1737,
```

```
            "low-warning": 123
        },
        "remote-media-rx-power-alarm-type": {
            "high-alarm": 2187,
            "low-alarm": 63,
            "high-warning": 1737,
            "low-warning": 79
        }
    },
    ...
    {
        "name": "fc/0/57",
        "identifier": "SFPDD",
        "connector": "SN",
        "transmission-type": "shortwave",
        "encoding": "PAM4",
        "vendor-name": "BROCADE",
        "vendor-oui": "00:05:1e",
        "part-number": "57-1000505-01",
        "vendor-revision": "1",
        "serial-number": "WAA12125C001096S",
        "date-code": "210729",
        "temperature": 37,
        "rx-power": 1005.00,
        "tx-power": 963.90,
        "current": 6.91,
        "voltage": 3298.00,
        "wavelength": 850,
        "power-on-time": 12895,
        "media-distance": {},
        "optical-cable-types": {},
        "media-optional-signal-value": "",
        "media-speed-capability": {
            "speed": [
                16,
                32,
                64
            ]
        },
        "media-speed-capability-v2": {
            "speed": [
                "16-gfc",
                "32-gfc",
                "64-gfc"
            ]
        },
        "nominal-baud-rate": 0,
        "baud-rate-maximum": 0,
        "baud-rate-minimum": 0,
        "fiber-9um-link-length": 0,
        "fiber-9m-link-length": 0,
        "multimode-fiber-om2-50um-length": 0,
        "multimode-fiber-om3-50um-length": 0,
```

```
                "multimode-fiber-om4-50um-length": 0,
                "multimode-fiber-625um-length": 0,
                "copper-link-length": 0,
                "enhanced-options": "0x0",
                "digital-diagnostics-type": "0x0",
                "media-optional-signals": {
                    "signal": [
                        "none"
                    ]
                },
                "status-control": "0x0",
                "electric-loopback-enabled": false,
                "optical-loopback-enabled": false,
                "alarm-flags": {
                    "high-alarm": "0x0",
                    "low-alarm": "0x0"
                },
                "warning-flags": {
                    "high-warning": "0x0",
                    "low-warning": "0x0"
                },
                "state-transitions": 1,
                "last-poll-time": "2023-03-01T14:50:00-07:00",
                "peer-data-available": false
            }
        ]
    }
}
```


brocade-name-server/fibrechannel-name-server

Description

The N_Port Identifier serves as the unique data base key for the

Name Server device list.

This list supports the following query parameter combinations:

port-id

port-name

node-name

Parameters

Table 132: brocade-name-server/fibrechannel-name-server

Name	Type	Description
port-id	fibrechannel:fcid-hex-string-type	A Fibre Channel address identifier assigned to an Nx_Port during either implicit or explicit fabric logins. The format of the port identifier is identical to the address identifier format. The port identifier serves as the unique data base key for the Name Server. This leaf can be queried.
port-name	fibrechannel:wwn-type	The device port world wide name (WWN). This leaf can be queried.
port-symbolic-name	string	The symbolic port name registered with the Name Server. It is subject to all the description and constraints of that object. The registered value displays within double quotes and is preceded by a numeric value indicating the length of the registered data. The numeric value is encapsulated within brackets. For example: [30] "dsim:brcdtgt:sw0[10.10.10.100]"
fabric-port-name	fibrechannel:wwn-type	The F_Port WWN to which the N_Port connects.
permanent-port-name	fibrechannel:wwn-type	The Name_Identifier associated with the physical Nx_Port. If there are multiple Name_Identifiers associated with a single physical Nx_Port via FDISC (see FC-FS) the permanent port name is the original Name_Identifier associated with that physical Nx_Port at fabric login.
node-name	fibrechannel:wwn-type	The device port world wide name (WWN). This leaf can be queried.
node-symbolic-name	string	The name registered with the Name Server as a Name Server symbolic node name object and is subject to all the description and constraints of that object. If not all Nx_Ports on the host bus adapter have the same node name, the node-symbolic-name resource is not returned for the host bus adapter. The registered value displays within double quotes and is preceded by a numeric value indicating the length of the registered data. The numeric value is encapsulated within brackets. For example: [30] "dsim:brcdtgt:sw0[10.10.10.100]"
class-of-service	fibrechannel:class-of-service-type	The Class of Service (CoS) levels supported by the device. A device can support multiple CoS levels.
fc4-type	fibrechannel:fc4-type-type	A set of supported data structure type values (for example, supported values of the type field in frame headers) for Device_Data and FC-4 Link_Data frames.
fc4-features	fibrechannel:fc4-features-type	The registered FC-4 features bits for a given FC-4 type.
port-type	fibrechannel:port-type-string-type	The port type of the device.

Name	Type	Description
state-change-registration	string	The state change registration (SCR) of the device. Returns an empty tag if the SCR value is not known. None: No registration. The device has not registered for any Remote State Change Notifications (RSCN) function. Fabric-Detected: Registered to receive all RSCN for events detected by the fabric. Nx-Port-Detected: Registered to receive all RSCN requests issued for events detected by the affected Nx_Port. Fabric-Name: Fabric Name change registration (FC Standard variant). Registered to receive an RSCN when the Fabric Principal switch changes. Peer-Zone: Registered to receive an RSCN when an associated Peer Zone changes. Brocade-Principal-Switch: Principal switch change registration (proprietary). Brocade-QoS: Access Gateway Quality-of-Service change registration (proprietary).
name-server-device-type	string	The definition for the type and role of a device. The device type can be Physical, Virtual, NPIV, or iSCSI. If the device type is not Physical, Virtual, NPIV, or iSCSI, the type displays as 'Undefined'. The role can be Initiator, Target, or Initiator+Target. If the device role is not registered, the role displays as 'Unknown'.
port-index	fibrechannel:user-port-number-type	The index number of the physical N_Port to which the device connects.
share-area	string	Indicates whether or not the port utilizes the Brocade Shared Area method of Fibre Channel addressing. yes : The port shares an area. no : The port does not share an area.
frame-redirection	string	Indicates whether or not the port is involved in Frame Redirection zoning. The device involved in frame redirection is specified as either 'virtual', 'host', or 'target'. yes : The port is involved in frame redirection. no : The port is not involved in frame redirection.
partial	string	Indicates whether or not the device entry is incomplete. Incomplete devices have routing established, but are not considered during device discovery (for example, during FC-GS Name Server queries). yes : The device entry is incomplete. no : The device entry is complete.
lsan	enumeration	Indicates whether or not the device is part of an active LSAN zone.
link-speed	string	Deprecated: Please reference the protocol-speed leaf instead. The link speed of the device (for example, 4G, 8G, 16G, and so on.) The link speed of end devices that log in to the edge fabric through F_Port trunk display the bandwidth of the F_Port trunk. For example, if an 8G host attached to AG login to the fabric through a 32G F_Port trunk, the device link speed for the 8G host is displayed as 32G.

Name	Type	Description
protocol-speed	fibrechannel:protocol-speed-type	The link speed (by protocol) of the device. The link speed of end devices that log in to the edge fabric through F_Port trunk display the aggregate bandwidth of the F_Port trunk. For example, if an 8-gfc host attached via an Access Gateway connected to the fabric through a 32-gfc F_Port Trunk, then device link speed for the 8-gfc host is displayed as 32-gfc.
port-properties	string	A list of proprietary port properties (for example, 'SIM Port' for Brocade SIM ports).
cascaded-ag	string	Indicates whether or not the device is connected via a cascaded Access Gateway (AG) connection. yes : The device is connected through a cascaded AG connection (for example, a 2-hop AG connection). no : The device is not connected through a cascaded AG connection.
connected-through-ag	string	Indicates whether or not the device is connected through AG. yes : The device is connected through AG. no : The device is not connected through AG.
real-device-behind-ag	string	Indicates whether or not the device is a virtualized device (NPIV) connected through AG. yes : The device is not a virtualized device (NPIV) connected through AG. no : The device is a virtualized device (NPIV) connected through AG.
fcoe-device	string	Indicates whether or not the device is an FCoE device. yes : The device is an FCoE device. no : The device is not an FCoE device.
slow-drain-device-quarantine	string	Indicates whether or not the device has been identified as having symptoms of credit stalling. yes : The device has been identified as having a credit stalled condition, and has its incoming traffic quarantined to a Low Priority VC. no : The device has not been flagged for having a credit stalled issue.
connected-through-fc-lag	boolean	Indicates whether or not the device is connected through FC-LAG. true : The device is connected through FC-LAG. false : The device is not connected through FC-LAG.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <fibrechannel-name-server>
  <port-id>10a00</port-id>
  <port-symbolic-name>[30] "dsim:brcdtgt:sw0[10.38.18.116]"</port-symbolic-name>
  <node-symbolic-name>[30] "dsim:brcdtgt:sw0[10.38.18.116]"</node-symbolic-name>
  <port-name>30:0a:c4:f5:7c:4e:59:54</port-name>
  <node-name>10:00:c4:f5:7c:4e:59:54</node-name>
  <port-index>10</port-index>
  <share-area> No </share-area>
  <frame-redirection> No </frame-redirection>
  <partial> No </partial>
  <lsan> No </lsan>
  <link-speed>16</link-speed>
  <state-change-registeration>0000000000</state-change-registeration>
  <port-type>N</port-type>
  <class-of-service>2,3</class-of-service>
  <fc4-type>FCP</fc4-type>
  <fabric-port-name>20:0a:c4:f5:7c:16:99:f4</fabric-port-name>
  <connected-through-ag>No</connected-through-ag>
  <port-properties/>
  <nameserver-devicetype>Physical Initiator</nameserver-devicetype>
  <real-device-behind-ag>No</real-device-behind-ag>
  <cascaded-ag>No</cascaded-ag>
  <slow-drain-device-quarantine>No</slow-drain-device-quarantine>
  <fcoe-device>No</fcoe-device>
  <fc4-features>FCP-Initiator</fc4-features>
  <permanent-port-name>30:03:00:05:33:a3:e8:85</permanent-port-name>
 </fibrechannel-name-server>
 <fibrechannel-name-server>
  <port-id>10b00</port-id>
  <port-symbolic-name>[30] "dsim:brcdtgt:sw0[10.38.18.116]"</port-symbolic-name>
  <node-symbolic-name>[30] "dsim:brcdtgt:sw0[10.38.18.116]"</node-symbolic-name>
  <port-name>30:0b:c4:f5:7c:4e:59:54</port-name>
  <node-name>10:00:c4:f5:7c:4e:59:54</node-name>
  <port-index>11</port-index>
  <share-area> No </share-area>
  <frame-redirection> No </frame-redirection>
  <partial> No </partial>
  <lsan> No </lsan>
  <link-speed>16</link-speed>
  <state-change-registeration>0000000000</state-change-registeration>
  <port-type>N</port-type>
  <class-of-service>2,3</class-of-service>
  <fc4-type>FCP </fc4-type>
  <fabric-port-name>20:0b:c4:f5:7c:16:99:f4</fabric-port-name>
  <connected-through-ag>No</connected-through-ag>
  <port-properties/>
  <nameserver-devicetype>Physical Initiator</nameserver-devicetype>
  <real-device-behind-ag>No</real-device-behind-ag>
```

```
   <cascaded-ag>No</cascaded-ag>
   <slow-drain-device-quarantine>No</slow-drain-device-quarantine>
   <fcoe-device>No</fcoe-device>
   <fc4-features>FCP-Initiator</fc4-features>
   <permanent-port-name>30:03:00:05:33:a3:e8:85</permanent-port-name>
  </fibrechannel-name-server>
 </Response>
```

brocade-security/acl-policy

Description

The security policies. This list is to get the security policies(SCC/FCS/DCC).
Each policy corresponds to a management method. The list of members of a policy acts as an access control list for that management method.
Before a policy is created, there is no enforcement for that management method, which is all access is granted. After a policy is created and a member is added to the policy, that policy is closed to all access except to included members. If all members are then deleted from the policy, all access is denied for that management access method.

Parameters

Table 133: brocade-security/acl-policy

Name	Type	Description
policy-activated	boolean	The current status of the policy (defined or active). true : Active. false: Defined.
policy-name	brocade-security-type:security-policy-name-type	The security policy name. Valid Values are - SCC_POLICY - FCS_POLICY - DCC_POLICY_* Policy name name must be capitalized. The policy name DCC_POLICY_* can be up to 30 characters, including the prefix DCC_POLICY_

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-security/active-dcc-policy-member-list

Description

Used for displaying members of the active DCC policy.

Parameters

Table 134: brocade-security/active-dcc-policy-member-list

Name	Type	Description
wwn	fibrechannel:wwn-type	The wwn member contains an 8 byte value that identifies the switch or node.
member-type	dcc-policy-member-type	Specifies the provided WWN member as device WWN or switch WWN.
domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
switch-name	string	A printable ASCII name for the switch.
ports	Container	Port Number list
- port	fibrechannel:user-port-number-type	Ports in the specified switch.
policy-name	brocade-security-type:security-policy-name-type	The security policy name. Valid Values are DCC_POLICY_* Policy name name must be capitalized. The policy name DCC_POLICY_* can be up to 30 characters, including the prefix DCC_POLICY_

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-security/active-fcs-policy-member-list

Description

Used for displaying members of the active FCS policy.

Parameters

Table 135: brocade-security/active-fcs-policy-member-list

Name	Type	Description
switch-wwn	fibrechannel:wwn-type	The switch-wwn member contains an 8 byte value that identifies the switch that contains the E_Port.
domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
switch-name	string	A printable ASCII name for the switch.
is-fcs-primary	boolean	Specifies whether the switch is primary or not. true : Primary. false: Non-Primary.
fcs-position	uint16	Specifies the position of the FCS Policy.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-security/active-scc-policy-member-list

Description

Used for displaying members of the active SCC policy.

Parameters

Table 136: brocade-security/active-scc-policy-member-list

Name	Type	Description
switch-wwn	fibrechannel:wwn-type	The switch-wwn member contains an 8 byte value that identifies the switch that contains the E_Port.
domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
switch-name	string	A printable ASCII name for the switch.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-security/auth-spec

Description

Replaces the configuration with the specified AAA service.

Parameters

Table 137: brocade-security/auth-spec

Name	Type	Description
authentication-mode	brocade-security-type:aaa-authspec-type	The authentication mode for RADIUS, TACACS+, and LDAP. MODE Descriptions local Default setting. Authenticates management connections against the local database only. radius Authenticates management connections against any RADIUS databases only. ldap Authenticates management connections against any LDAP databases only. tacacs+ Authenticates management connections against any TACACS+ databases only. radius;local Authenticates management connections against any RADIUS databases first. If RADIUS fails for any reason , authenticates against the local user database. radius;localbackup Authenticates management connections against any RADIUS databases. If RADIUS fails because the service is not available, it then authenticates against the local user database. This option directs the service to try the secondary authentication database only if the primary authentication database is not available. tacacs+;local Authenticates management connections against any TACACS+ databases first. If TACACS+ fails for any reason, it then authenticates against the local user database. tacacs+;localbackup Authenticates management connections against any TACACS+ databases first. If TACACS+ fails for any reason, it then authenticates against the local user database. This option directs the service to try the secondary authentication database only if the primary authentication database is not available. ldap;local Authenticates management connections against any LDAP databases first. If LDAP fails for any reason, it then authenticates against the local user database. ldap;localbackup Authenticates management connections against any LDAP databases first. If LDAP fails for any reason it then authenticates against the local user database. This option directs the service to try the secondary authentication database only if the primary authentication database is not available.
activate-no-log-out	boolean	Write-only leaf that you can set with a PATCH request; however, a GET request does not fetch this leaf. Whether a change in the authentication mode results in existing sessions being logged out automatically. true : There is no effect on existing sessions regardless of the chosen authentication mode. false: Terminates all existing sessions if the new authentication mode is one of the following: RADIUS only, LDAP only, TACACS+ only or Local.

Name	Type	Description
primary-auth-log-messages	boolean	Whether log messages for authentication failure display. true : Display log messages. false: Suppress log messages.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
 <?xml version="1.0"?>
<Response>
 <auth-spec>
  <authentication-mode>local</authentication-mode>
  <primary-auth-log-messages>true</primary-auth-log-messages>
 </auth-spec>
</Response>
```

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-security/authentication-configuration

Description

Displays the authentication configuration.

Equivalent to the CLI authutil.

Parameters

Table 138: brocade-security/authentication-configuration

Name	Type	Description
authentication-type	brocade-security-type:authentication-protocol-type	Specifies the available Authentication Protocols.
authentication-hash-type	brocade-security-type:authentication-hash-type	Specifies the available hash types.

Name	Type	Description
dh-group	brocade-security-type:dh-group-type	Specifies the available Authentication Group types.
switch-authentication-policy	brocade-security-type:authentication-policy-mode-type	Specifies the switch authentication policy
device-authentication-policy	brocade-security-type:authentication-policy-mode-type	Specifies the device authentication policy

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <authentication-configuration>
        <authentication-type>all</authentication-type>
        <authentication-hash-type>all</authentication-hash-type>
        <dh-group>dh-group-all</dh-group>
        <switch-authentication-policy>passive</switch-authentication-policy>
        <device-authentication-policy>off</device-authentication-policy>
    </authentication-configuration>
</Response>
```

brocade-security/defined-dcc-policy-member-list

Description

Shows defined DCC policies.

Only saved DCC policies will be there in a GET response. Any unsaved

DCC policies will not be present.

Each policy corresponds to a management method. The list of members of

a policy acts as an access control list for that management method.

Before a policy is created, there is no enforcement for that management

method, which is all access is granted. After a policy is created and a

member is added to the policy, that policy is closed to all access

except to included members. If all members are then deleted from the

policy, all access is denied for that management access method.

Note that only activated policies are enforced.

Parameters

Table 139: brocade-security/defined-dcc-policy-member-list

Name	Type	Description
wwn	fibrechannel:wwn-type	The wwn member contains an 8 byte value that identifies the switch or node.
member-type	dcc-policy-member-type	Specifies the provided WWN member as device WWN or switch WWN.
domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
switch-name	string	A printable ASCII name for the switch.
ports	Container	Port Number list
- port	fibrechannel:user-port-number-type	Ports in the specified switch.
policy-name	brocade-security-type:security-policy-name-type	The security policy name. Valid Values are - DCC_POLICY_* Policy name name must be capitalized. The policy name DCC_POLICY_* can be up to 30 characters, including the prefix DCC_POLICY_

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-security/defined-fcs-policy-member-list

Description

Displays defined FCS policy.

Only saved FCS policy will be present in a GET response. Any unsaved

FCS policy will not be present.

Note that only activated policies are enforced. When an FCS policy is

enabled in the switch, security policy management methods can be done

only through the Primary FCS switch.

Parameters

Table 140: brocade-security/defined-fcs-policy-member-list

Name	Type	Description
switch-wwn	fibrechannel:wwn-type	The switch-wwn member contains an 8 byte value that identifies the switch that contains the E_Port.
domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
switch-name	string	A printable ASCII name for the switch.
is-fcs-primary	boolean	Specifies whether the switch is primary or not. true : Primary. false: Non-Primary.
fcs-position	uint16	Specifies the position of the FCS Policy.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message and a “200 OK” status in the headers.

brocade-security/defined-scc-policy-member-list

Description

Displays defined SCC policy.

Only saved SCC policy will be there in a GET response. Any unsaved policies will not be present.

Each policy corresponds to a management method. The list of members of a policy acts as an access control list for that management method.

Before a policy is created, there is no enforcement for that management method, which is all access is granted. After a policy is created and a member is added to the policy, that policy is closed to all access except to included members. If all members are then deleted from the policy, all access is denied for that management access method.

Parameters

Table 141: brocade-security/defined-scc-policy-member-list

Name	Type	Description
switch-wwn	fibrechannel:wwn-type	The switch-wwn member contains an 8 byte value that identifies the switch that contains the E_Port.
domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
switch-name	string	A printable ASCII name for the switch.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-security/dh-chap-authentication-secret

Description

Displays the DH-CHAP shared secret key information from the database.

Parameters

Table 142: brocade-security/dh-chap-authentication-secret

Name	Type	Description
switch-wwn	fibrechannel:wwn-type	WWN for which a shared secret is configured.
domain-id	fibrechannel:domain-id-type	Domain ID for which a shared secret is configured.
switch-name	string	Switch name for which a shared secret is configured.
device-wwn	fibrechannel:wwn-type	Device WWN for which a shared secret is configured.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body and a “200 OK” status in the headers.

brocade-security/ipfilter-policy

Description

The IP filter policies. You use this container to manage the IP filter policies.

The IP filter policy sets up a packet filtering firewall to provide access control on the management IP interface. The IPv4 and IPv6 policies are either in the defined configuration or in the active configuration.

There can be maximum of eight IP filter policies on the switch, including the two default policies(default_ipv4 and default_ipv6).

The active policy must be the default policy or one of the policies in the defined configuration. Note that only active policies are enforced.

Parameters

Table 143: brocade-security/ipfilter-policy

Name	Type	Description
name	brocade-security-type:ipfilter-name-type	The IP filter policy name. The policy name can be up to 20 alphanumeric or underscore characters and must be unique. Note that policy names are case-insensitive and are always stored as lower case. The default_ipv4 and default_ipv6 names are reserved for the default IP filter policies.
ip-version	brocade-security-type:ipfilter-ip-version-type	The IP filter policy version type (IPv4 or IPv6).
is-policy-active	boolean	The current status of the policy (defined or active). true : Active. false: Defined.
is-default-policy	boolean	Whether the policy is a default or user-defined policy. true : Default policy. false: User-defined policy.

Name	Type	Description
action	brocade-security-type:ipfilter-action-type	Write-only leaf that you can set with PATCH request; however, a GET request does not fetch this leaf. The action to take on the IP filter policy. IP filter configurations are saved automatically after any change. Available actions include: o activate o clone
clone-destination-policy-name	brocade-security-type:ipfilter-name-type	Write-only leaf that you can set with PATCH request; however, a GET request does not fetch this leaf. The destination IP filter policy name for the clone operation. The policy name can be up to 20 alphanumeric or underscore characters and must be unique.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <ipfilter-policy>
  <name>default_ipv4</name>
  <ip-version>IPv4</ip-version>
  <is-policy-active>true</is-policy-active>
  <is-default-policy>true</is-default-policy>
 </ipfilter-policy>
 <ipfilter-policy>
  <name>default_ipv6</name>
  <ip-version>IPv6</ip-version>
  <is-policy-active>true</is-policy-active>
  <is-default-policy>true</is-default-policy>
 </ipfilter-policy>
 <ipfilter-policy>
  <name>fwd_ipv4</name>
  <ip-version>IPv4</ip-version>
  <is-policy-active>false</is-policy-active>
  <is-default-policy>false</is-default-policy>
 </ipfilter-policy>
 <ipfilter-policy>
  <name>46RB7ATw0XsERUO2X5h</name>
  <ip-version>IPv4</ip-version>
```

```
   <is-policy-active>false</is-policy-active>
   <is-default-policy>false</is-default-policy>
  </ipfilter-policy>
 </Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-security/ipfilter-rule

Description

The IP filter rule configurations.

Parameters

Table 144: brocade-security/ipfilter-rule

Name	Type	Description
policy-name	brocade-security-type:ipfilter-name-type	The IP filter policy name. The policy name can be up to 20 alphanumeric or underscore characters and must be unique. Note that policy names are case-insensitive and are always stored as lower case. The default_ipv4 and default_ipv6 names are reserved for the default IP filter policies.
index	uint16	The position of the IP filter entry in the IP filter table. The rule number must be between 1 and the current maximum rule number plus one.
source-ip	brocade-security-type:host-type	The source IP address. For IPv4 filters, the address must be a 32-bit address in dot notation or a CIDR-style IPv4 prefix. For IPv6 filters, the address must be a 12-bit IPv6 address in any format specified by RFC3513 or a CIDR-style IPv6 prefix. Note that the source IP option is optional for FORWARD traffic. If the source IP leaf is not specified, the leaf is automatically given the default value 'any'.
destination-start-port	inet:port-number	The destination port number or the starting destination port number for a port number range. The blocking or permitting of port from 1 through 65535 is allowed.

Name	Type	Description
destination-end-port	inet:port-number	The ending destination port number of the port number range. If no destination-end-port is specified during IP filter rule creation (POST), the destination end port is equal to destination start port.
protocol	brocade-security-type:ipfilter-protocol-type	The protocol type (tcp or udp).
permission	brocade-security-type:ipfilter-permission-type	The permit or deny action associated with this rule. You can permit or deny (block)ports from 1 through 65535.
traffic-type	brocade-security-type:ipfilter-traffic-type	The type of traffic that allowed for the specified IP address. The input traffic type manages traffic on IP management interfaces. The forward traffic type manages bidirectional traffic between the external Ethernet interface (eth0/bond0) and the inband management interface (inbd+).
destination-ip	brocade-security-type:host-type	The destination IP address. For IPv4 filters, the address must be a 32-bit address in dot notation or a CIDR-style IPv4 prefix. For IPv6 filters, the address must be a 12-bit IPv6 address in any format specified by RFC3513 or a CIDR-style IPv6 prefix. Note that the destination IP address is optional for INPUT traffic. If the destination IP leaf is not specified, the leaf is automatically given the default value 'any'.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <ipfilter-rule>
  <policy-name>default_ipv4</policy-name>
  <index>1</index>
  <source-ip>any</source-ip>
  <destination-start-port>22</destination-start-port>
  <destination-end-port>22</destination-end-port>
  <protocol>tcp</protocol>
  <permission>permit</permission>
  <destination-ip>any</destination-ip>
 </ipfilter-rule>
...
```

```
  <ipfilter-rule>
   <policy-name>46RB7ATw0XsERUO2X5h</policy-name>
   <index>1</index>
   <source-ip>177.171.183.217/30</source-ip>
   <destination-start-port>2</destination-start-port>
   <destination-end-port>2</destination-end-port>
   <protocol>udp</protocol>
   <permission>permit</permission>
   <destination-ip>243.112.59.242/0</destination-ip>
  </ipfilter-rule>
 </Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-security/ldap-role-map

Description

A list of the LDAP AD server role to default switch role mappings.

Used to map a Lightweight Directory Access Protocol (LDAP)

Active Directory (AD) server role to one of the default

roles available on a switch.

Parameters

Table 145: brocade-security/ldap-role-map

Name	Type	Description
home-virtual-fabric	home-virtual-fabric-type	Specifies the account's home Virtual Fabric. If the value is not configured in ldap-role-map, GET will fetch '0' value.
chassis-access-role	user-config-role-type	Specifies the account's access permissions regarding chassis-level commands.
ldap-role	string	The LDAP role to be mapped to a switch role. The role must be a valid AD server role. The maximum length is 64 printable ASCII characters. The following characters are not allowed '/[]:;|=,+*?<>'.

Name	Type	Description
switch-role	string	The default switch role to be mapped to an LDAP role. The format is either a string containing switch user type or switch user type with virtual fabric role ID information separated by ';'. Example strings include: o user o user=1-10;admin=11-128

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <ldap-role-map>
  <ldap-role>User1</ldap-role>
  <switch-role>admin=1-128</switch-role>
  <home-virtual-fabric>1</home-virtual-fabric>
  <chassis-access-role/>
 </ldap-role-map>
</Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-security/ldap-server

Description

The LDAP server configuration.

Parameters

Table 146: brocade-security/ldap-server

Name	Type	Description
server	inet:host	The IP address or a server name in dot-decimal notation. If the specified server IP address or name already exists in the current configuration, the command fails and generates an error. However, the command does not validate the server name against the IP address in the configuration. Make sure to avoid duplicate configuration of the same server, one specified by the name, the other specified by the IP address
port	inet:port-number	The LDAP server port number.
domain	inet:domain-name	The name of the active directory domain. For example, brocade.com
timeout	brocade-security-type:aaa-timeout-type	The response timeout for the LDAP server.
tls-mode	brocade-security-type:ldap-tls-mode-type	The configuration that defines whether to use StartTLS or LDAPS. starttls : use StartTLS ldaps : use LDAPS
position	uint16	The position to which you want to move the LDAP server. If the server position is set to 2 (old position was 5), the server is moved to position 2 and any existing server at position 2 is moved to position 3.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
 <ldap-server>
  <server>223.134.45.70</server>
  <port>38970</port>
  <domain>brocadcom.com</domain>
  <timeout>25</timeout>
  <tls-mode>starttls</tls-mode>
  <position>1</position>
 </ldap-server>
</Response>
```

DELETE Request Response

When the DELETE operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

POST Request Response

When the POST operation is successful, the response contains an empty message body and a “201 Created” status appears in the header.

brocade-security/management-rbac-map

Description

This is used for listing all the Role Based Access Control (RBAC)

permission with context type details for the URLs.

Parameters

Table 147: brocade-security/management-rbac-map

Name	Type	Description
url	string	The URL path for the resource.
rbac-class	brocade-security-type:rbac-class-type	The Role Based Access Control (RBAC) permission used for the URL.
context-type	brocade-security-type:rbac-context-type	Context type of the URL.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <management-rbac-map>
        <url>/rest/auth-token</url>
        <rbac-class>none</rbac-class>
```

```
        <context-type>both</context-type>
    </management-rbac-map>
...
    <management-rbac-map>
        <url>/rest/running/brocade-zone/zone</url>
        <rbac-class>Zoning</rbac-class>
        <context-type>vf</context-type>
    </management-rbac-map>
</Response>
```

brocade-security/password

Description

The user account password container.

Parameters

Table 148: brocade-security/password

Name	Type	Description
user-name	brocade-security-type:user-config-user-name-type	The user name. User names are case-sensitive and can contain up to 32 alphanumeric characters, including periods (.) and underscores (_). Note that user name is mandatory parameter for changing the password.
old-password	brocade-security-type:user-password-type	The old password. The password must be encoded with the Base64 encoding scheme.
new-password	brocade-security-type:user-password-type	The new password. The password must be encoded with the Base64 encoding scheme. Note that new password is mandatory parameter for changing the password.

Produces

- application/xml
- application/json

Supported Methods

PATCH

Example Response

PATCH Request Response

When the PATCH operation is successful, the response contains an empty message body and a “204 No Content” status appears in the header.

brocade-security/password-cfg

Description

The generic password policy configuration for all user accounts.

Parameters

Table 149: brocade-security/password-cfg

Name	Type	Description
minimum-length	int32	The minimum length of the password. The minimum can be set between 8 and 40 characters.
character-set	int32	The minimum length on the character set (upper and lowercase letters and special characters) to use in the password. The value must be less than or equal to minimum-length leaf.
user-name-allowed	boolean	The validation check to determine if the username is used in the password. If activated, the username (in both forward and reverse direction) cannot be used in the password. For example, if username is 'testuser', then the password 'testuser123' or 'resutset567' is not allowed. true : Activated. false: Deactivated.
minimum-lower-case-character	int32	The minimum number of lowercase alphabetic characters that must appear in the password. This value must be less than or equal to the minimum-length leaf.
minimum-upper-case-character	int32	The minimum number of uppercase alphabetic characters that must appear in the password. This value must be less than or equal to the minimum-length leaf.
minimum-numeric-character	int32	The minimum number of numeric digits that must appear in the password. This value must be less than or equal to the minimum-length leaf.
minimum-special-character	int32	The minimum number of punctuation characters that must appear in the password. This value must be less than or equal to the minimum-length leaf. All displayable, non-alphanumeric punctuation characters, except the colon (:), are allowed.
past-password-history	int32	The number of past password values that are disallowed when setting a new password.
minimum-password-age	int32	The minimum number of days that must elapse before a password can be changed. When the maximum-password-age leaf is set to non-zero value, the minimum-password-age value must be set to a value less than or equal to the maximum-password-age value.
maximum-password-age	int32	The maximum number of days that can elapse before a password must be changed. This is the password expiration period. Setting this parameter to 0 disables password expiration.
warn-on-expire	int32	The number of days prior to password expiration that a warning about password expiration displays.
lock-out-threshold	int32	The number of times a user can specify an incorrect password during login before the account is locked. The number of failed login attempts is counted from the last successful login. Setting this parameter to 0 disables the lockout mechanism.

Name	Type	Description
lock-out-duration	int32	The time after which a previously locked account automatically unlocks. Setting this to 0 disables lockout duration, and requires an administrative action to unlock the account.
admin-lock-out-enabled	boolean	Whether the lockout policy is enabled for the admin account. true : Enabled. false: Disabled.
admin-lock-out-console-access-enabled	boolean	Allow console access for admin account when admin lockout is enabled. admin-lock-out-console-access-enabled is applicable only when admin-lock-out-enabled is set to true. When admin-lock-out-enabled is set to false state, admin-lock-out-console-access-enabled will be automatically set to false state and this leaf will be skipped in the GET response. true : Enabled. false: Disabled.
repeat-character-limit	int32	The length of repeated character sequences that are disallowed. For example, if the 'repeat' value is set to 3, a password 'passAAAword' is disallowed because it contains the repeated sequence 'AAA'. However, a password 'passAAword' is allowed because no repeated character sequence exceeds two characters.
sequence-character-limit	int32	The length of sequential character sequences that are disallowed. In a character sequence, the ASCII value of each contiguous character differs by one. The ASCII value for the characters in the sequence must all be increasing or decreasing. For example, if the 'sequence' value is set to 3, a password 'passABCword' is disallowed because it contains the sequence 'ABC'.
password-config-changed	boolean	The account password policy status. true : Configuration changed. false: Default configuration.
reverse-user-name-allowed	boolean	The validation check to determine whether the password is an exact reverse string of the username. true : Activated. false: Deactivated.
hash-type	brocade-security-type:hash-algorithm-type	The hash type (md5, sha256, or sha512).
manual-hash-enabled	boolean	A password change due to hash type change can be manual and is not enforced at login. true : Enabled. false: Disabled.
enforce-expire	boolean	Write-only leaf that you can set with a PATCH request; however, a GET request does not fetch this leaf. The expiration of the password for all users. If activated, the user is prompted for a password change on the next successful login. true : Activated. false: Deactivated.
minimum-difference	uint16	The number of character differences expected between the old and new password.

Name	Type	Description
password-action	brocade-security-type:password-cfg-operation-type	Write-only leaf that you can set with a PATCH request; however a GET request does not fetch this leaf. The configuration or deletion of password policies. Valid actions include: o hash-config o default o delete-all

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <password-cfg>
        <hash-type>sha512</hash-type>
        <manual-hash-enabled>false</manual-hash-enabled>
        <minimum-length>8</minimum-length>
        <character-set>0</character-set>
        <user-name-allowed>true</user-name-allowed>
        <minimum-lower-case-character>0</minimum-lower-case-character>
        <minimum-upper-case-character>0</minimum-upper-case-character>
        <minimum-numeric-character>0</minimum-numeric-character>
        <minimum-special-character>0</minimum-special-character>
        <past-password-history>1</past-password-history>
        <minimum-password-age>0</minimum-password-age>
        <maximum-password-age>0</maximum-password-age>
        <warn-on-expire>0</warn-on-expire>
        <lock-out-threshold>0</lock-out-threshold>
        <lock-out-duration>30</lock-out-duration>
        <admin-lock-out-enabled>false</admin-lock-out-enabled>
        <repeat-character-limit>1</repeat-character-limit>
        <sequence-character-limit>1</sequence-character-limit>
        <password-config-changed>false</password-config-changed>
        <reverse-user-name-allowed>false</reverse-user-name-allowed>
        <minimum-difference>0</minimum-difference>
    </password-cfg>
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-security/policy-distribution-config

Description

Manages the fabric data distribution configuration parameters.

Parameters

Table 150: brocade-security/policy-distribution-config

Name	Type	Description
scc-distribute-permission	brocade-security-type:security-policy-distribute-permission-type	Switch can be locally configured to allow or reject SCC policy.
dcc-distribute-permission	brocade-security-type:security-policy-distribute-permission-type	Switch can be locally configured to allow or reject DCC policy.
password-distribute-permission	brocade-security-type:security-policy-distribute-permission-type	Switch can be locally configured to allow or reject password policy.
fcs-distribute-permission	brocade-security-type:security-policy-distribute-permission-type	Switch can be locally configured to allow or reject FCS policy.
auth-distribute-permission	brocade-security-type:security-policy-distribute-permission-type	Switch can be locally configured to allow or reject authentication policy.
ip-filter-distribute-permission	brocade-security-type:security-policy-distribute-permission-type	Switch can be locally configured to allow or reject IP filter policy.
fabric-wide-policies	Container	List of security databases for fabric wide consistency.
- policy	brocade-security-type:fabric-wide-consistency-policy-type	Automatic distribution of a fabric-wide consistency policy is limited to SCC, DCC, and FCS policies. Policies can be enforced fabric-wide in tolerant or strict mode.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
 <?xml version="1.0"?>
<Response>
    <policy-distribution-config>
        <scc-distribute-permission>accept</scc-distribute-permission>
        <dcc-distribute-permission>accept</dcc-distribute-permission>
        <password-distribute-permission>accept</password-distribute-permission>
        <fcs-distribute-permission>accept</fcs-distribute-permission>
        <auth-distribute-permission>accept</auth-distribute-permission>
        <ip-filter-distribute-permission>accept</ip-filter-distribute-permission>
        <fabric-wide-policies/>
    </policy-distribution-config>
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-security/radius-server

Description

The RADIUS server configuration.

Parameters

Table 151: brocade-security/radius-server

Name	Type	Description
server	inet:host	The IP address or server name in dot-decimal notation. If the specified server IP address or name already exists in the current configuration, the command fails and generates an error. However, the command does not validate the server name against the IP address in the configuration. Make sure to avoid duplicate configuration of the same server, one specified by the name, the other specified by the IP address
port	inet:port-number	The RADIUS server port number.
secret	string	Write-only leaf that you can set with a PATCH or post request; however, a GET request does not fetch this leaf. the common secret between the switch and the RADIUS server. The secret is mandatory in a POST request and must be encoded with the Base64 encoding scheme.
timeout	brocade-security-type:aaa-timeout-type	The response timeout for the RADIUS server.

Name	Type	Description
authentication	union {brocade-security-type:aaa-protocols-type;string;}	The remote authentication protocol for the RADIUS server. Valid protocols are, o pap o chap o peap-mschapv2
encryption-type	brocade-security-type:aaa-encryption-algorithm-type	The encryption algorithm type for the RADIUS server. Valid algorithm include: o none o aes256
position	uint16	The position to which you want to move the RADIUS server. For example, if you change the server position to 2 (old position was 5), the server is moved to position 2 and any existing server at position 2 is moved to position 3.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body, and a “200 OK” status in the headers.

DELETE Request Response

When the DELETE request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

POST Request Response

When the POST request is successful, the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

brocade-security/rbac-class

Description

Information about role-based access control (RBAC) permissions for

meta-object format (MOF) class(es) and for CLI command(s).

This list supports queries against class-name and command-name

leaf objects.

This list supports the following query parameter combinations:

class-name

command-name

class-name + command-name

Parameters

Table 152: brocade-security/rbac-class

Name	Type	Description
class-name	brocade-security-type:rbac-class-type	Specifies RBAC class name. This leaf can be queried
command-name	string	Specifies CLI command name. This leaf can be queried
command-option	string	Specifies sub-option for the command.
permission	brocade-security-type:rbac-permission-type	Specifies RBAC permission information
context	brocade-security-type:rbac-context-type	Specifies the context

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <rbac-class>
        <class-name>AG</class-name>
        <command-name>ag</command-name>
        <command-option>addwwnfailovermapping</command-option>
        <permission>OM</permission>
    </rbac-class>
    <rbac-class>
        <class-name>AG</class-name>
        <command-name>ag</command-name>
        <command-option>addwwnmapping</command-option>
        <permission>OM</permission>
    </rbac-class>
```

```
...
    <rbac-class>
        <class-name>Zoning</class-name>
        <command-name>zoneshow</command-name>
        <command-option>transdiffs</command-option>
        <permission>O</permission>
    </rbac-class>
    <rbac-class>
        <class-name>Zoning</class-name>
        <command-name>zoneshow</command-name>
        <command-option>transdiffsonly</command-option>
        <permission>O</permission>
    </rbac-class>
</Response>
```

brocade-security/role-config

Description

Used to create or modify user-defined roles,to define permissions
for these roles based on role-based access control (RBAC) permissions
and meta-object format (MOF) classes, and to display the configured
roles. Two types of access control restriction exist in Fabric OS:

Restriction by MOF class: A MOF class groups similar Fabric OS commands
into feature sets that share the same access permissions.
By assigning one or more MOF classes to a role, the account with the
specified role can access all the commands included in these classes.

Restriction by RBAC access level: Restrict access by setting RBAC
one of the following access levels for the role.
The RBAC permissions are set per class.
– O = observe
– OM = observe-modify

Parameters

Table 153: brocade-security/role-config

Name	Type	Description
name	brocade-security-type:role-name-type	The name of the role to be created or modified. Role name must be unique; it is case-insensitive and can contain only alpha characters. The maximum number of user-defined roles allowed in FOS is 64.
role-description	string	Specifies a description for the role. Colons (:) are not permitted.

Name	Type	Description
role-type	brocade-security-type:rbac-role-type	Specifies whether the role is Default or User-defined.
class-permissions	Container	Specifies one or more MOF classes and its RBAC permission to which the role should have access.
- class-permission	brocade-security-type:rbac-class-permission-type	Specifies RBAC class name with '=' seperated RBAC class permission. The maximum length allowed for class name is 64 characters. Valid RBAC permissions are 'OM' and 'O'. NOTE: ==== All RBAC classes with no permission settings are skipped in the GET response for a role. Example : LocalUserEnvironment=O Security=OM

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <role-config>
        <name>admin</name>
        <role-description>Administrator</role-description>
        <role-type>system-defined</role-type>
        <class-permissions>
            <class-permission>AdminDomains=OM</class-permission>
            <class-permission>ADSelect=OM</class-permission>
            <class-permission>AG=OM</class-permission>
            <class-permission>Audit=OM</class-permission>
            <class-permission>Authentication=OM</class-permission>
            <class-permission>Blade=OM</class-permission>
            <class-permission>ChassisConfiguration=OM</class-permission>
            <class-permission>ChassisManagement=OM</class-permission>
            <class-permission>ConfigManagement=OM</class-permission>
            <class-permission>Configure=OM</class-permission>
            <class-permission>DCE=OM</class-permission>
            <class-permission>Diagnostics=OM</class-permission>
            <class-permission>DMM=OM</class-permission>
            <class-permission>EncryptionConfiguration=OM</class-permission>
            <class-permission>EncryptionManagement=OM</class-permission>
```

```
            <class-permission>EthernetConfig=OM</class-permission>
            <class-permission>Fabric=OM</class-permission>
            <class-permission>FabricDistribution=OM</class-permission>
            <class-permission>FabricRouting=OM</class-permission>
            <class-permission>FCoE=OM</class-permission>
            <class-permission>FICON=OM</class-permission>
            <class-permission>FIPSCfg=OM</class-permission>
            <class-permission>FirmwareKeyManagement=OM</class-permission>
            <class-permission>FirmwareManagement=OM</class-permission>
            <class-permission>FRUManagement=OM</class-permission>
            <class-permission>HA=OM</class-permission>
            <class-permission>IPfilter=OM</class-permission>
            <class-permission>ISCSI=OM</class-permission>
            <class-permission>LayerTwo=OM</class-permission>
            <class-permission>License=OM</class-permission>
            <class-permission>LocalUserEnvironment=OM</class-permission>
            <class-permission>Logging=OM</class-permission>
            <class-permission>LogSupportsave=OM</class-permission>
            <class-permission>ManagementAccessConfiguration=OM</class-permission>
            <class-permission>ManagementServer=OM</class-permission>
            <class-permission>MAPS=OM</class-permission>
            <class-permission>NameServer=OM</class-permission>
            <class-permission>Nocheck=OM</class-permission>
            <class-permission>NxPortManagement=OM</class-permission>
            <class-permission>PhysicalComputerSystem=O</class-permission>
            <class-permission>PKI=OM</class-permission>
            <class-permission>PortMirror=OM</class-permission>
            <class-permission>RADIUS=OM</class-permission>
            <class-permission>Reboot=OM</class-permission>
            <class-permission>Restricted=OM</class-permission>
            <class-permission>RoleConfig=OM</class-permission>
            <class-permission>RoutingAdvanced=OM</class-permission>
            <class-permission>RoutingBasic=OM</class-permission>
            <class-permission>Security=OM</class-permission>
            <class-permission>SessionManagement=OM</class-permission>
            <class-permission>SNMP=OM</class-permission>
            <class-permission>SRM=OM</class-permission>
            <class-permission>Statistics=OM</class-permission>
            <class-permission>StatisticsDevice=OM</class-permission>
            <class-permission>StatisticsPort=OM</class-permission>
            <class-permission>SwitchConfiguration=OM</class-permission>
            <class-permission>SwitchManagement=OM</class-permission>
            <class-permission>SwitchManagementIPConfiguration=OM</class-permission>
            <class-permission>SwitchPortConfiguration=OM</class-permission>
            <class-permission>SwitchPortSecurityConfiguration=OM</class-permission>
            <class-permission>SwitchPortManagement=OM</class-permission>
            <class-permission>Topology=OM</class-permission>
            <class-permission>USBManagement=OM</class-permission>
            <class-permission>UserManagement=OM</class-permission>
            <class-permission>WWNCard=OM</class-permission>
            <class-permission>Zoning=OM</class-permission>
        </class-permissions>
    </role-config>
```

```
...
    <role-config>
        <name>operator</name>
        <role-description>Operator</role-description>
        <role-type>system-defined</role-type>
        <class-permissions>
            <class-permission>ADSelect=OM</class-permission>
            <class-permission>AG=OM</class-permission>
            <class-permission>Blade=OM</class-permission>
            <class-permission>ChassisConfiguration=OM</class-permission>
            <class-permission>ChassisManagement=OM</class-permission>
            <class-permission>ConfigManagement=O</class-permission>
            <class-permission>Configure=O</class-permission>
            <class-permission>Diagnostics=OM</class-permission>
            <class-permission>EthernetConfig=O</class-permission>
            <class-permission>Fabric=O</class-permission>
            <class-permission>FabricRouting=O</class-permission>
            <class-permission>FCoE=OM</class-permission>
            <class-permission>FICON=OM</class-permission>
            <class-permission>FirmwareKeyManagement=O</class-permission>
            <class-permission>FirmwareManagement=OM</class-permission>
            <class-permission>FRUManagement=OM</class-permission>
            <class-permission>HA=O</class-permission>
            <class-permission>ISCSI=O</class-permission>
            <class-permission>LayerTwo=OM</class-permission>
            <class-permission>License=OM</class-permission>
            <class-permission>LocalUserEnvironment=OM</class-permission>
            <class-permission>Logging=OM</class-permission>
            <class-permission>LogSupportsave=OM</class-permission>
            <class-permission>ManagementAccessConfiguration=OM</class-permission>
            <class-permission>ManagementServer=OM</class-permission>
            <class-permission>MAPS=OM</class-permission>
            <class-permission>NameServer=O</class-permission>
            <class-permission>Nocheck=OM</class-permission>
            <class-permission>NxPortManagement=OM</class-permission>
            <class-permission>PhysicalComputerSystem=O</class-permission>
            <class-permission>PKI=O</class-permission>
            <class-permission>Reboot=OM</class-permission>
            <class-permission>RoutingAdvanced=O</class-permission>
            <class-permission>RoutingBasic=OM</class-permission>
            <class-permission>SessionManagement=OM</class-permission>
            <class-permission>SNMP=O</class-permission>
            <class-permission>Statistics=OM</class-permission>
            <class-permission>StatisticsDevice=OM</class-permission>
            <class-permission>StatisticsPort=OM</class-permission>
            <class-permission>SwitchConfiguration=OM</class-permission>
            <class-permission>SwitchManagement=OM</class-permission>
            <class-permission>SwitchManagementIPConfiguration=OM</class-permission>
            <class-permission>SwitchPortConfiguration=OM</class-permission>
            <class-permission>SwitchPortManagement=OM</class-permission>
            <class-permission>Topology=O</class-permission>
            <class-permission>WWNCard=OM</class-permission>
            <class-permission>Zoning=O</class-permission>
```

```
        </class-permissions>
    </role-config>
</Response>
```

DELETE Request Response

When the DELETE request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

POST Request Response

When the POST request is successful, the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

brocade-security/sec-crypto-cfg

Description

The active sec-crypto configurations.

Parameters

Table 154: brocade-security/sec-crypto-cfg

Name	Type	Description
ssh-cipher	brocade-security-type:default-string-type	The active SSH cipher configuration.
ssh-kex	brocade-security-type:default-string-type	The active SSH kex configuration.
ssh-mac	brocade-security-type:default-string-type	The active SSH mac configuration.
https-cipher	brocade-security-type:default-string-type	The active HTTPS cipher configuration.
https-tlsv13-cipher	brocade-security-type:default-string-type	The active HTTPS TLSv1.3 cipher configuration.
radius-cipher	brocade-security-type:default-string-type	The active RADIUS cipher configuration.
ldap-cipher	brocade-security-type:default-string-type	The active LDAP cipher configuration.
syslog-cipher	brocade-security-type:default-string-type	The active Syslog cipher configuration.
https-tls-protocol	brocade-security-type:tls-protocol-type	The active HTTPS TLS protocol.
radius-tls-protocol	brocade-security-type:tls-protocol-type	The active RADIUS TLS protocol.

Name	Type	Description
ldap-tls-protocol	brocade-security-type:tls-protocol-type	The active LDAP TLS protocol.
syslog-tls-protocol	brocade-security-type:tls-protocol-type	The active Syslog TLS protocol.
x509v3-validation-mode	brocade-security-type:validation-mode-type	The X509v3 certificate validation mode type. X509v3 certificate validation permits Fabric OS to enable or disable certificate validation during certificate import and session establishment. The following identifies the differences between Basic mode and Strict mode for X509v3 certificate validation. Basic mode: o 'Certificate Sign' is not mandatory for 'KeyUsage' in a CA certificate. o OCSP validation for identity certificates is not performed. o 'BasicConstraints' field is not mandatory in a CA certificate. o 'CA:True' value is not mandatory in a CA certificate. o 'Digital Signature' is not mandatory for 'KeyUsage' in a identity certificate. o Audit logs for establishing and terminating TLS session are not printed. o SSH sessions with a switch, the 1GB data transfer limit is not enforced for rekeying. o For RADIUS, LDAP, or Syslog secure authentication using a certificate, a hostname mismatch error is ignored. o For RADIUS, LDAP, or Syslog secure authentication using a certificate, hostname validation is not performed. o For RADIUS, LDAP, or Syslog secure authentication over TLS using a certificate, if there is a hostname mismatch error, no audit log entry is generated. Strict mode: o 'Certificate Sign' is mandatory for 'KeyUsage' in a CA certificate. o OCSP validation for identity certificates is performed. o 'BasicConstraints' field is mandatory in a CA certificate. o 'CA:True' value is mandatory in a CA certificate. o 'Digital Signature' is mandatory for 'KeyUsage' in a identity certificate. o Audit log for establishing and terminating TLS session is printed for each TLS session established. In a switch this applies both as client and as server. o In SSH sessions with a switch, the 1GB data transfer limit is enforced for rekeying. o For RADIUS, LDAP, or Syslog secure authentication using a certificate, a hostname mismatch error causes authentication failure o For RADIUS, LDAP, or Syslog secure authentication, hostname validation is enforced and a hostname mismatch error causes authentication failure (hostname resolution failure also causes authentication failure). o For RADIUS, LDAP, or Syslog secure authentication over TLS using a certificate, if there is a hostname mismatch error, an audit log entry is generated.

Name	Type	Description
crypto-version	brocade-security-type:crypto-version-type	Crypto Version number.
boot-up-self-test-enabled	boolean	State of boot-up Self Tests true : Enabled. false: Disabled.
boot-up-self-test-status	brocade-security-type:boot-up-self-test-status-type	Status of boot-up Self Tests execution

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <sec-crypto-cfg>
        <ssh-cipher>aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc</ssh-cipher>
        <ssh-kex>ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-
sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1</ssh-kex>
        <ssh-mac>hmac-sha1,hmac-sha2-256,hmac-sha2-512</ssh-mac>
        <https-cipher>ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4</https-cipher>
        <https-tlsv13-cipher>TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256</
https-tlsv13-cipher>
        <radius-cipher>ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4</radius-cipher>
        <ldap-cipher>ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4</ldap-cipher>
        <syslog-cipher>ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4</syslog-cipher>
        <https-tls-protocol>any</https-tls-protocol>
        <radius-tls-protocol>any</radius-tls-protocol>
        <ldap-tls-protocol>any</ldap-tls-protocol>
        <syslog-tls-protocol>any</syslog-tls-protocol>
        <x509v3-validation-mode>basic</x509v3-validation-mode>
        <crypto-version>9.1.0</crypto-version>
        <boot-up-self-test-enabled>true</boot-up-self-test-enabled>
        <boot-up-self-test-status>pass</boot-up-self-test-status>
    </sec-crypto-cfg>
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-security/sec-crypto-cfg-template

Description

A list of available templates.

Parameters

Table 155: brocade-security/sec-crypto-cfg-template

Name	Type	Description
name	brocade-security-type:sec-crypto-cfg-template-name-type	The template name.
template	string	The content of templates.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <sec-crypto-cfg-template>
        <name>default_fips</name>
        <template>[Ver]  0.3
[SSH]
Enc:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
Kex:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256
Mac:hmac-sha1,hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
LDAP_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
RAD_Protocol:TLSv1.2
LDAP_Protocol:TLSv1.2
[LOG]
Syslog_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
```

```
Syslog_Protocol:TLSv1.2
[HTTPS]
Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
Protocol:TLSv1.2
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[X509v3]
Validation:Basic
[Compliance]
BootUpSelfTests:yes
Zeroize:yes
FIPSInside:yes
SelfTests:default
CryptoIntegrityTest:yes
FIPSExtn:yes
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_generic</name>
        <template>[Ver]  0.2
[SSH]
Enc:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
Kex:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-
hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1
Mac:hmac-sha1,hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4
LDAP_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4
RAD_Protocol:Any
LDAP_Protocol:Any
[LOG]
Syslog_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4
Syslog_Protocol:Any
[HTTPS]
Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4
Protocol:Any
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[X509v3]
Validation:Basic
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_cc</name>
        <template>[Ver]  0.3
[SSH]
Enc:aes128-cbc,aes256-cbc
Kex:diffie-hellman-group14-sha1,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521
Mac:hmac-sha1,hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
```

```
LDAP_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
RAD_Protocol:TLSv1.2
LDAP_Protocol:TLSv1.2
[LOG]
Syslog_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
Syslog_Protocol:TLSv1.2
[HTTPS]
Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!3DES:!aNULL
Protocol:TLSv1.2
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[X509v3]
Validation:Strict
[Compliance]
BootUpSelfTests:yes
Zeroize:no
FIPSInside:no
SelfTests:default
CryptoIntegrityTest:yes
FIPSExtn:no
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_strong</name>
        <template>[Ver]  0.2
[SSH]
Enc:aes128-ctr,aes192-ctr,aes256-ctr
Kex:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-
hellman-group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,curve25519-sha256
Mac:hmac-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4:!AEAD:!SHA1:!AESGCM
LDAP_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4:!AEAD:!SHA1:!AESGCM
RAD_Protocol:TLSv1.2
LDAP_Protocol:TLSv1.2
[LOG]
Syslog_Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!
CHACHA20:!SEED:!RC4:!AEAD:!SHA1:!AESGCM
Syslog_Protocol:TLSv1.2
[HTTPS]
Ciphers:ECDSA:ECDH:RSA:AES:3DES:!RSAPSK:!DHEPSK:!PSK:!DSS:!AESCCM8:!AESCCM:!ARIAGCM:!CAMELLIA:!CHACHA20:!
SEED:!RC4:!AEAD:!SHA1:!AESGCM
Protocol:TLSv1.3
Ciphers_tlsv1.3:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
[X509v3]
Validation:Basic
</template>
    </sec-crypto-cfg-template>
</Response>
```

brocade-security/sec-crypto-cfg-template-action

Description

Allows you to perform actions on the crypto template files.

You cannot overwrite or delete the default templates; however, they can be uploaded, edited, and then downloaded with a different name.

You can use the default or user defined templates to applying or verifying crypto configurations.

Parameters

Table 156: brocade-security/sec-crypto-cfg-template-action

Name	Type	Description
remote-user-name	string	Specifies the user name for the host.
remote-host-ip	inet:host	Specifies the remote host ip address.
remote-user-password	string	Password for the remote user. Password must be encoded with Base64 encoding scheme
remote-directory	string	Specifies the remote directory fully qualified path name(FQPN). If action is import, remote directory must be fully qualified file name. If the action is export, remote directory can be fully qualified file name (to export template with a name of user's choice) or directory name (to export template with the name same as that of template on switch).
file-transfer-protocol-type	sec-crypto-cfg-file-transfer-protocol-type	Specifies the protocol as either SCP, SFTP, or FTP for file transfer
template-name	brocade-security-type:sec-crypto-cfg-template-name-type	The template name.
action	brocade-security-type:sec-crypto-cfg-actions-type	The operation to perform. Valid operations include: o apply o verify o import o export The response should be empty content with status code 204 when the requested action is successful.

Produces

- application/xml
- application/json

Supported Methods

DELETE, PATCH

Example Response

DELETE Request Response

When the DELETE request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-security/security-certificate-action

Description

The import or export action of third-party certificates on a switch,

including Public Key Infrastructure (PKI) based certificates,

Lightweight Directory Access Protocol (LDAP) certificates, RADIUS

certificates, syslog CA certificates and extension certificates.

Parameters

Table 157: brocade-security/security-certificate-action

Name	Type	Description
remote-user-name	string	Specifies the user name for the host.
remote-host-ip	inet:host	Specifies the remote host ip address.
remote-user-password	string	Password for the remote user. Password must be encoded with Base64 encoding scheme
remote-directory	string	Specifies the remote directory fully qualified path name(FQPN).
protocol	brocade-security-type:seccertmgmt-protocol-type	The protocol type for import or export. Example: scp or ftp.
certificate-entity	brocade-security-type:certificate-entity-type	The certificate entity (CSR, switch, CA, CA Client, or Server certificate).
certificate-type	brocade-security-type:certificate-application-type	The certificate type (commoncert, https, radius, ldap, syslog, fcap or extension).
certificate-name	string	The certificate name.
ca-certificate	string	The preimported local CA certificate name.
operation	brocade-security-type:seccertmgmt-operation-type	The security certificate operations (such as import or export).
keypair-tag	string	The keypair tag is a user-defined name used to identify a local certificate uniquely.

Produces

- application/xml
- application/json

Supported Methods

DELETE, PATCH

Example Response

DELETE Request Response

When the DELETE request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-snmp/access-control

Description

The SNMP access control list is to restrict SNMP GET,

SET, and trap operations to the hosts under an host subnet area.

Deprecated : Please use ipfilter-policy list instead.

Parameters

Table 158: brocade-snmp/access-control

Name	Type	Description
index	uint16	The label for this object.
host	union {inet:ip-address;inet:ip-prefix;}	The subnet area of the access host. The IP address, for which SNMP operations works only for the hosts configured in ACL list. The IP address supports both IPv4 and IPv6 addresses.
access-level	snmp-types:access-permission	The access level of the SNMP access control entry. The access level can be either read-only or read-write. If the access host is patched with default value then the access level also will be set with default value (read-write) whether the input is either read-only or read-write.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <access-control>
        <index>0</index>
        <host>0.0.0.0</host>
        <access-level>rw</access-level>
    </access-control>
    <access-control>
        <index>1</index>
        <host>0.0.0.0</host>
        <access-level>rw</access-level>
    </access-control>
...
    </access-control>
    <access-control>
        <index>5</index>
        <host>0.0.0.0</host>
        <access-level>rw</access-level>
    </access-control>
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-snmp/mib-capability

Description

The MIBs or traps to enable or disable in

the MIB capability.

Parameters

Table 159: brocade-snmp/mib-capability

Name	Type	Description
mib-name	snmp-types:mibs-name	The MIB name.
is-mib-enabled-state	boolean	Indicates whether MIB is enabled or disabled. true : Enabled. false: Disabled.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <mib-capability>
        <mib-name>FE-MIB</mib-name>
        <is-mib-enabled-state>true</is-mib-enabled-state>
    </mib-capability>
    <mib-capability>
        <mib-name>SW-MIB</mib-name>
        <is-mib-enabled-state>true</is-mib-enabled-state>
    </mib-capability>
...
    <mib-capability>
        <mib-name>T11-FC-ZONE-SERVER-MIB</mib-name>
        <is-mib-enabled-state>false</is-mib-enabled-state>
    </mib-capability>
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-snmp/system

Description

The system-wide SNMP configurations

Parameters

Table 160: brocade-snmp/system

Name	Type	Description
description	string	A printable ASCII string that describes the entity. The parameter should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters. Refer to RFC 1213.
location	string	The physical location of this node. Refer to RFC 1213.
contact	string	The contact person for this managed node, together with how to contact this person. Refer to RFC 1213.

Name	Type	Description
informs-enabled	boolean	Indicates whether informs is enabled or disabled in the switch. true : Enabled. false: Disabled.
encryption-enabled	boolean	Indicates whether the password encryption flag is enabled or disabled. If password encryption is enabled, the authentication and privacy passwords are stored as encrypted. If password encryption is disabled, the authentication and privacy passwords are stored as plain text. If the authentication and privacy protocol is configured then the corresponding password should be configured with more than 8 characters. true : Enabled. false: Disabled.
audit-interval	uint16	The SNMP Audit interval.
security-get-level	snmp-types:security-level	The SNMP security access level for the GET operation. Deprecated: Please use 'security-get-level-string' leaf instead.
security-set-level	snmp-types:security-level	The SNMP security access level for the SET operation. Deprecated: Please use 'security-set-level-string' leaf instead.
security-get-level-string	snmp-types:security-level-string-type	The SNMP security access level for the GET operation. Modifying the security access level to No Access for GET operation might impact the SNMP GET / SNMP walk query triggered by the applications monitoring the system.
security-set-level-string	snmp-types:security-level-string-type	The SNMP security access level for the SET operation.
snmpv1-enabled	boolean	Indicates whether snmpv1 is enabled or disabled in the switch. true : Enabled. false: Disabled.
default-config	Container	A list of default SNMP configurations.
- default-control	snmp-types:default-control-type	This list is a write only container that is utilized to default SNMP configuration (such as snmpv1, snmpv3, access control, system group, mib capability, and audit interval).

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
```

```
<Response>
    <system>
        <description>Fibre Channel Switch.</description>
        <location>End User Premise.</location>
        <contact>Field Support.</contact>
        <informs-enabled>true</informs-enabled>
        <security-get-level>0</security-get-level>
        <security-set-level>3</security-set-level>
        <security-get-level-string>no-security</security-get-level-string>
        <security-set-level-string>no-access</security-set-level-string>
        <audit-interval>60</audit-interval>
        <encryption-enabled>false</encryption-enabled>
        <snmpv1-enabled>true</snmpv1-enabled>
    </system>
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-snmp/trap-capability

Description

A list of traps to enable or disable in the MIB capability.

The traps t11ZsRequestRejectNotify,

t11ZsMergeFailureNotify, t11ZsMergeSuccessNotify,

t11ZsDefZoneChangeNotify and t11ZsActivateNotify

are read-only and write operation for MIB

capability is not supported.

Parameters

Table 161: brocade-snmp/trap-capability

Name	Type	Description
trap-name	snmp-types:traps-name	The trap name.
is-trap-enabled-state	boolean	Indicates whether the trap is enabled or disabled. true : Enabled. false: Disabled.
severity	snmp-types:severity-level	The trap recipient severity level. When an event occurs and its severity level is at or below the set value (none, critical, error, warning, information, and debug), the Event Trap traps (swEventTrap) are sent to configured trap recipients.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <trap-capability>
        <trap-name>swFCPortScn</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swEventTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
...
   <trap-capability>
        <trap-name>t11ZsActivateNotify</trap-name>
        <is-trap-enabled-state>false</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-snmp/v1-account

Description

The SNMPv1 user account to access the system resource via SNMP.

It also contains the snmpv1 host recipients to receive the

SNMPv1 traps. Refer to RFC 1157.

Parameters

Table 162: brocade-snmp/v1-account

Name	Type	Description
index	uint16	The label for this object.
community-name	string	The community name.
community-group	snmp-types:group-name	Indicates whether the SNMPv1 community belongs to read-only or read-write group. The community group is mandatory in case of POST operation to create a v1 account.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <v1-account>
        <index>1</index>
        <community-group>read-write</community-group>
        <community-name>Secret C0de</community-name>
    </v1-account>
    <v1-account>
        <index>2</index>
        <community-group>read-write</community-group>
        <community-name>OrigEquipMfr</community-name>
    </v1-account>
...
   <v1-account>
        <index>6</index>
        <community-group>read-only</community-group>
        <community-name>FibreChannel</community-name>
    </v1-account>
</Response>
```

DELETE Request Response

When the DELETE request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

POST Request Response

When the POST request is successful, the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

brocade-snmp/v1-trap

Description

The SNMPv1 trap notification. Refer to RFC 3413.

Parameters

Table 163: brocade-snmp/v1-trap

Name	Type	Description
index	uint16	The label for this object.
host	union {inet:host;inet:ip-prefix;}	The IP address of the trap recipient system.
trap-severity-level	snmp-types:severity-level	The trap recipient severity level. When an event occurs and its severity level is at or below the set value (none, critical, error, warning, information, and debug), the Event Trap traps (swEventTrap) are sent to configured trap recipients.
port-number	inet:port-number	The UDP port where SNMP traps are received.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <v1-trap>
        <index>1</index>
        <host>0.0.0.0</host>
        <trap-severity-level>none</trap-severity-level>
        <port-number>162</port-number>
    </v1-trap>
```

```
    <v1-trap>
        <index>2</index>
        <host>0.0.0.0</host>
        <trap-severity-level>none</trap-severity-level>
        <port-number>162</port-number>
    </v1-trap>
...
    <v1-trap>
        <index>6</index>
        <host>0.0.0.0</host>
        <trap-severity-level>none</trap-severity-level>
        <port-number>162</port-number>
    </v1-trap>
</Response>S
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-snmp/v3-account

Description

The SNMPv3 user account. This parameter is used to accesses system
via SNMPv3 in a secured manner by means of authentication
and privacy. This parameter is also used to receive the traps and
informs notifications for the configured host recipient.
Refer to RFC 3414.

Parameters

Table 164: brocade-snmp/v3-account

Name	Type	Description
index	uint16	The label for this object.
user-name	string	The name of the user that connects to the agent.
user-group	snmp-types:group-name	Indicates whether the SNMPv3 user belongs to a read-only or a read-write group. The user group is mandatory in case of POST operation to create a v3 account.
authentication-protocol	snmp-types:authentication-protocol-type	The authorization protocol(MD5, SHA or SHA512) for the SNMPv3 user. Refer to RFC 3414.
privacy-protocol	snmp-types:privacy-protocol-type	The privacy protocol (DES, AES-128 or AES-256) for the SNMPv3 user. Refer to RFC 3414. The privacy protocol cannot be changed from nopriv if the authentication-protocol is noAuth.

Name	Type	Description
authentication-password	string	This is write-only leaf and it defines the authentication password for the SNMPv3 user to access the system resources. The password should be base64 encoded. Refer to RFC 3414. The password cannot be set if the authentication-protocol is noAuth. The password is mandatory to set the authentication-protocol other than noAuth.
privacy-password	string	This is write-only leaf and it defines the privacy password for the SNMPv3 user to access the system resources. The password should be base64 encoded. Refer to RFC 3414. The password cannot be set if the privacy-protocol is noPriv or if the authentication-protocol is noAuth. The password is mandatory to set the privacy-protocol other than noPriv.
manager-engine-id	string	The user-defined engine ID for the SNMP manager which is used to receive the SNMPv3 informs notifications. This parameter is applicable only when informs is enabled.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <v3-account>
        <index>1</index>
        <user-name>admin</user-name>
        <user-group>read-write</user-group>
        <authentication-protocol>sha</authentication-protocol>
        <privacy-protocol>des</privacy-protocol>
        <authentication-password/>
        <privacy-password/>
        <manager-engine-id>00:00:00:00:00:00:00:00:00:00</manager-engine-id>
    </v3-account>
    <v3-account>
        <index>2</index>
        <user-name>snmpadmin2</user-name>
```

```
        <user-group>read-write</user-group>
        <authentication-protocol>noauth</authentication-protocol>
        <privacy-protocol>nopriv</privacy-protocol>
        <authentication-password/>
        <privacy-password/>
        <manager-engine-id>11:11:11:11:11:11:11:11:11:11:12</manager-engine-id>
    </v3-account>
...
    <v3-account>
        <index>7</index>
        <user-name>snmpadmin7</user-name>
        <user-group>read-write</user-group>
        <authentication-protocol>noauth</authentication-protocol>
        <privacy-protocol>nopriv</privacy-protocol>
        <authentication-password/>
        <privacy-password/>
        <manager-engine-id>11:11:11:11:11:11:11:11:11:11:12</manager-engine-id>
    </v3-account>
</Response>
```

DELETE Request Response

When the DELETE request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

POST Request Response

When the POST request is successful, the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

brocade-snmp/v3-trap

Description

the SNMPv3 trap notification. Refer to RFC 3413.

Parameters

Table 165: brocade-snmp/v3-trap

Name	Type	Description
trap-index	uint16	The label for this object.
usm-index	uint16	The index of the user account associated with the trap recipient index.
host	union {inet:host;inet:ip-prefix;}	The IP address of the trap recipient system.

Name	Type	Description
trap-severity-level	snmp-types:severity-level	The trap recipient severity level. When an event occurs and its severity level is at or below the set value none, critical, error, warning, informational, and debug)the Event Trap traps (swEventTrap) are sent to the configured trap recipients.
port-number	inet:port-number	The UDP port number where SNMP traps are received.
informs-enabled	boolean	Indicates whether the informs is enabled or disabled. true : Enabled. false: Disabled.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <v3-trap>
        <trap-index>1</trap-index>
        <usm-index>1</usm-index>
        <host>0.0.0.0</host>
        <port-number>162</port-number>
        <trap-severity-level>none</trap-severity-level>
        <informs-enabled>false</informs-enabled>
    </v3-trap>
    <v3-trap>
        <trap-index>2</trap-index>
        <usm-index>2</usm-index>
        <host>0.0.0.0</host>
        <port-number>162</port-number>
        <trap-severity-level>none</trap-severity-level>
        <informs-enabled>false</informs-enabled>
    </v3-trap>
 ...
    <v3-trap>
        <trap-index>6</trap-index>
        <usm-index>6</usm-index>
        <host>0.0.0.0</host>
        <port-number>162</port-number>
        <trap-severity-level>none</trap-severity-level>
        <informs-enabled>false</informs-enabled>
    </v3-trap>
```

```
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-supportlink/supportlink-history

Description

The supportLink history is used to retrieve the failures

occurred during the inventory upload operations.

Parameters

Table 166: brocade-supportlink/supportlink-history

Name	Type	Description
index	uint16	The index of the specific upload failure entry and it is used as a label for this object.
time-stamp	yang:date-and-time	The time when supportLink upload operation failed.
server	inet:host	SupportLink destination server address or domain name.
port	inet:port-number	The HTTPS port number of the destination server.
error-code	int32	It indicates the exact failure occurred during supportLink upload operation.
error-message	string	The error message provides the reason for the supportLink upload operation failures.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <supportlink-history>
        <index>1</index>
        <time-stamp>2021-09-30T07:21:00-06:00</time-stamp>
        <server>asc-gateway.brm.bsnlab.broadcom.net</server>
        <port>8030</port>
```

```
        <error-code>500</error-code>
        <error-message>Unknown error</error-message>
    </supportlink-history>
    <supportlink-history>
        <index>2</index>
        <time-stamp>2021-09-30T06:59:06-06:00</time-stamp>
        <server>asc-gateway.brm.bsnlab.broadcom.net</server>
        <port>8030</port>
        <error-code>500</error-code>
        <error-message>Unknown error</error-message>
    </supportlink-history>
...
    <supportlink-history>
        <index>20</index>
        <time-stamp>2021-09-30T03:59:00-06:00</time-stamp>
        <server>asc-gateway.brm.bsnlab.broadcom.net</server>
        <port>8030</port>
        <error-code>500</error-code>
        <error-message>Unknown error</error-message>
    </supportlink-history>
</Response>
```

brocade-supportlink/supportlink-profile

Description

The SupportLink profile is used to specify the configuration that need to be collected/uploaded to the destination server.

Parameters

Table 167: brocade-supportlink/supportlink-profile

Name	Type	Description
supportlink-enabled	boolean	Specifies whether SupportLink feature is enabled or not. true : Indicates that automatic transfer of support data to Broadcom has been enabled. By enabling this function the admin agrees that data, which is not intended to transfer personal data, however personal data could be included, will be automatically transferred to Broadcom. false : Indicates that automatic transfer of support data to Broadcom has been disabled.
server	inet:host	Destination server address or domain name.
port	inet:port-number	The HTTPS port number of the destination server.
user-name	union {string;string;}	The existing user name or an email address configured on the server.
start-date	union {enumeration;string;}	The start date is in weekday [Mon, Tue, Wed, Thu, Fri, Sat, Sun] or exact date [MM/DD/YYYY] format.

Name	Type	Description
period	uint32	The frequency in which the service must be triggered. For example, if Sunday is configured for start date, the period must be set to 7 days to make it as weekly runs. If the exact date is configured, for example 07/30/2018, the period must be set to any value as per the maximum and minimum timer configuration.
start-time	uint32	The start time of the support data upload. Specified exact time in 24 hour format. For example, 1 means 01:00 AM.
end-time-period	uint32	The maximum time period from the start time of the upload service. Actual upload time is random time computed by device at every reboot, between start time and end time.
random-start-time	string	The random upload time that is calculated from start time and end time period of support data service.
retry-time	uint32	The upload service starts again after an hour(s) elapsed from the failure, if a non-zero value is set.
collection-time	string	The collection time of the support data. Specified in HH:MM in 24 hour format.
proxy-user	string	The username setup on the proxy server. This parameter is required only in proxy server environment.
proxy-password	brocade-security-type:base64-string-type	The proxy server password for the user. The password should be base64 encoded. Refer to RFC 3414. This parameter is required only in proxy server environment.
proxy-server	union {inet:host;fibrechannel:brocade-empty-string-type;}	The proxy server address or domain name. This parameter is required only in proxy server environment. Response doesn't include proxy-server attribute, if it is not configured.
proxy-port	inet:port-number	The proxy server port number. This parameter is required only in proxy server environment. Response doesn't include proxy-port attribute, if it is not configured.
proxy-protocol	enumeration	The protocols used to communicate with proxy server. This parameter is required only in proxy server environment. Response doesn't include proxy-protocol attribute, if it is not configured.
group-tag	string	The organizational group of switches.
next-collection-time	union {yang:date-and-time;string;}	The next support data collection time in the switch.
next-service-start-time	union {yang:date-and-time;string;}	The next support data upload time to the server.
next-service-retry-time	yang:date-and-time	The next support data upload retry time to the server.
last-collection-time	yang:date-and-time	The last successful support data collection time in the switch.
last-upload-time	yang:date-and-time	The last successful support data upload time to the server.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <supportlink-profile>
        <supportlink-enabled>true</supportlink-enabled>
        <server>asc-gateway.brm.bsnlab.broadcom.net</server>
        <port>8030</port>
        <user-name>j.doe@company.com</user-name>
        <period>1</period>
        <start-time>21</start-time>
        <start-date>05/08/2019</start-date>
        <collection-time>4:16</collection-time>
        <retry-time>0</retry-time>
        <end-time-period>12</end-time-period>
        <random-start-time>05/09/2019 4:16</random-start-time>
        <next-collection-time>2021-11-06T04:16:00-06:00</next-collection-time>
        <next-service-start-time>2021-11-06T04:16:00-06:00</next-service-start-time>
        <proxy-user/>
        <group-tag>sqa_rest</group-tag>
    </supportlink-profile>
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-time/clock-server

Description

The Network Time Protocol (NTP) IP address(es) or LOCL (local).

Parameters

Table 168: brocade-time/clock-server

Name	Type	Description
active-server	brocade-time-type:ts-ntp-type	The IP address of the active server or LOCL (for the local server).
ts-auth-spec	brocade-time-type:ts-authspec-type	The authspec mode: symmetric or noauth
ts-legacy-mode	boolean	The legacy mode: enable or disable

Name	Type	Description
ntp-server-address	Container	The NTP server IP address(es) or LOCL (for the local server).
- server-address	brocade-time-type:ts-ntp-type	A list of the NTP server IP addresses or LOCL (for the local server). Configure an NTP server IP address to set the clock server IP address and enable fabric-wide clock synchronization with the specified clock server. You can configure up to 8 NTP servers.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <clock-server>
        <ntp-server-address>
            <server-address>10.10.10.10</server-address>
        </ntp-server-address>
        <active-server>NONE</active-server>
        <ts-auth-spec>noauth</ts-auth-spec>
        <ts-legacy-mode>false</ts-legacy-mode>
    </clock-server>
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-time/ntp-clock-server

Description

A list of Network Time Protocol (NTP) IP address(es) and associated symmetric key index(es). You can configure up to 8 NTP servers and symmetric key indexes.

Parameters

Table 169: brocade-time/ntp-clock-server

Name	Type	Description
server	ts-ntp-type	NTP server IP address or LOCL (for the local server). Configure an NTP server IP address to set the clock server IP address and enable fabric-wide clock synchronization with the specified clock server.
index	int32	NTP symmetric key configured for the NTP server. Configure an NTP key index to be associated with the NTP server for symmetric encryption. Valid indexes are from 1 to 65535. -1 will be returned when index is not in use.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <ntp-clock-server>
        <server>10.10.10.10</server>
        <index>-1</index>
    </ntp-clock-server>
</Response>
```

brocade-time/ntp-clock-server-key

Description

The Network Time Protocol (NTP) Symmetric Keys.

You can configure up to 256 symmetric keys.

Parameters

Table 170: brocade-time/ntp-clock-server-key

Name	Type	Description
index	int32	The symmetric key index.
type	brocade-time-type:ts-key-type	The symmetric key type.
key	string	The symmetric key can be in one of two formats: 1) Exactly 20 characters in length: any ASCII character allowed 2) 21 characters or more in length: only hex characters allowed This leaf will only be displayed for an account with admin role.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <ntp-clock-server-key>
        <index>1</index>
        <type>MD5</type>
        <key>password90123456789012345678 90</key>
    </ntp-clock-server-key>
    <ntp-clock-server-key>
        <index>2</index>
        <type>HMAC-SHA1</type>
        <key>password9012345678903456789 0</key>
    </ntp-clock-server-key>
    <ntp-clock-server-key>
        <index>3</index>
        <type>HMAC-SHA256</type>
        <key>password901234567890123456789123456789012345678 90</key>
    </ntp-clock-server-key>
    <ntp-clock-server-key>
        <index>4</index>
        <type>CMAC-AES-128</type>
        <key>password90123456789012901234567890</key>
    </ntp-clock-server-key>
```

```
</Response>
```

DELETE Request Response

When the DELETE request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

POST Request Response

When the POST request is successful, the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

brocade-time/time-zone

Description

The container to configure the system time zone name or the

offset (hours,min or hours format).

Note that a change to the System Time Zone takes effect at next reboot.

On a chassis, both CPs reboot simultaneously.

Parameters

Table 171: brocade-time/time-zone

Name	Type	Description
name	brocade-time-type:ts-timezone-type	The name of a time zone from the time zone database (for example, Europe/Andorra).
gmt-offset-hours	int16	The number of hours relative to GMT.
gmt-offset-minutes	int16	The number of minutes relative to the hour offset.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <time-zone>
        <name>America/Denver</name>
    </time-zone>
```

```
</Response>
```

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

brocade-traffic-optimizer/partial-performance-group

Description

This list the parially functional performance group

because the Performance Group is not present in all

the switches in the fabric

Parameters

Table 172: brocade-traffic-optimizer/partial-performance-group

Name	Type	Description
name	performance-group-id-type	Traffic Optimizer performance group Name.
reason	string	Reason for partially functional
domains	Container	List of domains not associated with this Performance Group
- domain	fibrechannel:domain-id-type	Domain not associated with this Performance Group

Produces

- application/xml
- application/json

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "partial-performance-group": [
            {
                "name": "PG_SCSI_32G",
                "reason": "TO Incapable Platforms",
                "domains": {
                    "domain": [
                        43,
                        92,
                        235,
                        236,
                        237
```

```
                    ]
                }
            },
            {
                "name": "PG_SCSI_32G",
                "reason": "TO Profile Mismatch",
                "domains": {
                    "domain": [
                        234
                    ]
                }
            },
            {
                "name": "PG_SCSI_64G",
                "reason": "TO Incapable Platforms",
                "domains": {
                    "domain": [
                        43,
                        92,
                        235,
                        236,
                        237
                    ]
                }
            }
        ]
    }
}
```


brocade-traffic-optimizer/performance-group

Description

A Performance Group (PG) is a high level software construct to group the traffic flows based on predefined attributes, such as the flows destination speed, frame size, priority, protocols, etc. This list the PG associated with the active profile

Performance Group statistics will be skipped if number of active flows exceeded the limit for the number of flows monitored by the flow vision or number of flow supported by Traffic optimizer for this plarform and FOS version

Parameters

Table 173: brocade-traffic-optimizer/performance-group

Name	Type	Description
identifier	performance-group-id-type	Performance Group name to identify PG
stats-duration	enumeration	The elapsed time the statistics for a performance group taken
io-category	enumeration	Read or write or aggregated read and write statistics for a performance group
performance-group-description	string	A printable ASCII string that describes the Performance Group entity.
aggregate-throughput	yang:gauge64	Aggregated throughput counter on PG
aggregate-iops	yang:gauge64	Aggregated I/O operation counter on PG
max-io-latency	yang:gauge64	Maximum I/O Latency counter on PG
io-errors	yang:gauge64	I/O errors counter on PG
flow-count	yang:gauge64	Number of flows associated with the PG
aggregate-violations	yang:gauge64	High Latency flow count on PG
virtual-channels	Container	List of virtual channel assigned to this performance group
- virtual-channel	virtual-channel-type	VC number associated with PG

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following and a “200 OK” status in the headers.

```
{
    "Response": {
        "performance-group-profile": [
            {
                "name": "Sys_TrafOpt_Version1",
                "profile-activated": true,
                "profile-scheduled-to-activate": false,
                "profile-description": "This System profile contains Performance Groups that are created based
 on device speed",
                "performance-groups": {
                    "identifier": [
                        "PG_4G_8G_16G",
                        "PG_SDDQ_QOS_LOW_1",
```

```
                        "PG_SDDQ_QOS_LOW_2",
                        "PG_QOS_HIGH_1",
                        "PG_QOS_HIGH_2",
                        "PG_QOS_HIGH_3",
                        "PG_QOS_HIGH_4",
                        "PG_QOS_HIGH_5",
                        "PG_SCSI_32G",
                        "PG_SCSI_64G"
                    ]
                }
            },
            {
                "name": "Sys_TrafOpt_Version2",
                "profile-activated": false,
                "profile-scheduled-to-activate": false,
                "profile-description": "This System profile contains Performance Groups that are created based
 on device speed, protocol and congestion state",
                "performance-groups": {
                    "identifier": [
                        "PG_4G_8G_16G",
                        ...
                        "PG_NVME_64G"
                    ]
                }
            },
            {
                "name": "TrafOpt_Custom",
                "profile-activated": false,
                "profile-scheduled-to-activate": false,
                "profile-description": "This custom profile contains Performance Groups that are created based
 on user inputs on the fabric environment like speed and protocol.",
                "performance-groups": {
                    "identifier": [
                        "PG_DEFAULT",
                        ...
                        "PG_NVME_64G"
                    ]
                },
                "configured-speed-protocol-types": {},
                "pending-speed-protocol-types": {
                    "pending-speed-protocol-type": [
                        "NVMe-64G",
                        ...
                        "SCSI-64G"
                    ]
                }
            }
        ]
    }
}
```

brocade-traffic-optimizer/performance-group-flows

Description

This lists all the flows and associated PGs

Parameters

Table 174: brocade-traffic-optimizer/performance-group-flows

Name	Type	Description
identifier	performance-group-id-type	The Performance Group ID
source-pid	fibrechannel:fcid-hex-string-type	The source PID of the flow
destination-pid	fibrechannel:fcid-hex-string-type	The destination PID of the flow
virtual-channel	virtual-channel-type	VC number associated with the flow

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body, and a “200 OK” status in the headers.

brocade-traffic-optimizer/performance-group-profile

Description

The list of profile to configured in the

Traffic optimizer

Parameters

Table 175: brocade-traffic-optimizer/performance-group-profile

Name	Type	Description
name	string	Traffic Optimizer profile Name.
profile-activated	boolean	Indicate whether profile is active or not. true: Active false: InActive
profile-scheduled-to-activate	boolean	Indicate whether profile is scheduled to activate. true: Scheduled false: Not Scheduled
delay-days	uint32	Time to activate the scheduled profile in days
delay-minutes	uint32	Time to activate the scheduled profile in minutes on top of delay-days value
profile-description	string	A printable ASCII string that describes the performace group Profile.
performance-groups	Container	List of performance group assigned to the profile
- identifier	performance-group-id-type	The Performance Group name identifier
configured-speed-protocol-types	Container	List of speed and protocol combinations associated with this custom profile. This will be listed when TrafOpt_custom profile is active.
- configured-speed-protocol-type	performance-group-speed-protocol-type	Speed and protocol configured in the custom profile like SCSI-32G, SCSI-16G. etc
pending-speed-protocol-types	Container	This list the speed and protocol combinations associated with the custom profile that is not active. These pending configurations will become active on subsequent custom profile activation.
- pending-speed-protocol-type	performance-group-speed-protocol-type	Pending speed and protocol configuration associated with the custom profile like SCSI-32G, SCSI-16G. etc

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body, and a “200 OK” status in the headers.

brocade-usb/usb-file

Description

To display files or folders for corresponding folder name

Parameters

Table 176: brocade-usb/usb-file

Name	Type	Description
path	operation-types:usb-path-type	Path to the file.
name	operation-types:usb-filename-type	Name of file in the USB device.
size	uint32	The size of the file, in bytes.
date	yang:date-and-time	The creation date of the file.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <usb-file>
    <path>/usb/usbstorage</path>
    <name>FOS-823540</name>
    <date>2020-12-31T06:54:46+00:00</date>
    <size>189980</size>
  </usb-file>
  <usb-file>
    <path>/usb/usbstorage/config</path>
    <name>config_export.txt</name>
    <date>2021-02-15T13:43:14+00:00</date>
    <size>267724</size>
  </usb-file>
  <usb-file>
    <path>/usb/usbstorage/defects</path>
    <name>hasm_logs</name>
```

```
    <date>2021-02-19T06:23:58+00:00</date>
    <size>117488</size>
  </usb-file>
 <Response>
```

brocade-zone/defined-configuration

Description

The defined zoning configuration.

Parameters

Table 177: brocade-zone/defined-configuration

Name	Type	Description
cfg [cfg-name]	List	The name of the zone configuration.
- cfg-name	zoning-name-type	The zone configuration name.
- member-zone	Container	A list of zone members in the configuration.
-- zone-name	zoning-name-type	The zone name.
zone [zone-name]	List	A list of the members in the zone. The members can be identified only as a WWN, domain,index, or a zone alias.
- zone-name	zoning-name-type	The zone name.
- zone-type	zone-type-type	The zone type. Note that target-created peer zone (TDZ) types cannot be created or modified (only deleted). Also note that this field must be supplied for any user-created peer zone operation. Deprecated: For write operations, use the 'zone-type-string' leaf instead. For read operations, this leaf is returned in responses with valid data comparable to the 'zone-type-string' leaf. Write operations performed on the zone-type leaf in the absence of a zone member-entry are not supported.
- zone-type-string	zone-type-string-type	The zone type. Note that target-created peer zone (TDZ) types cannot be created or modified (only deleted). Also note that this field must be supplied for any user-created peer zone operation. In the event of a write operation containing both the deprecated 'zone-type' leaf as well as the 'zone-type-string' leaf, both leafs must convey the same type. If they do not convey the same type, then the write operation results in an error. Write operations performed on the zone-type-string leaf in the absence of a zone member-entry are not supported.
- member-entry	Container	The zone member.

Name	Type	Description
-- entry-name	zone-member-type	A list of the members in the zone. The members can be identified only as a WWN, domain and index, or zone alias. If the zone-type or zone-type-string leaf field(s) specify a regular zone, then the entry-name list field is required. Otherwise, this entry-name list is optional for user-created peer zones.
-- principal-entry-name	zone-member-type	A list of the principal members in the peer zone. The members can be identified only as a WWN, domain,index, or zone alias.
alias [alias-name]	List	A list of zone aliases.
- alias-name	zoning-name-type	The zone alias name.
- member-entry	Container	The alias member.
-- alias-entry-name	union {fibrechannel:wwn-type;domain-index-type;}	The zone alias. The members are either a port WWN or domain,index.

Produces

- application/xml
- application/json

Supported Methods

DELETE, GET, HEAD, POST, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <defined-configuration>
        <cfg>
            <cfg-name>cfg20</cfg-name>
            <member-zone>
                <zone-name>68161_EMC</zone-name>
                <zone-name>68163_EMC</zone-name>
                <zone-name>68164_EMC</zone-name>
                <zone-name>68164_NETAPP</zone-name>
                <zone-name>ct_zone07</zone-name>
            </member-zone>
        </cfg>
...
        <cfg>
            <cfg-name>default_backup</cfg-name>
            <member-zone>
                <zone-name>68161_EMC</zone-name>
                <zone-name>68163_EMC</zone-name>
                <zone-name>68164_EMC</zone-name>
```

```
                <zone-name>68164_NETAPP</zone-name>
                <zone-name>68165_EMC1</zone-name>
                <zone-name>68165_EMC2</zone-name>
                <zone-name>68165_EMC3</zone-name>
                <zone-name>68165_NETAPP1</zone-name>
                <zone-name>68165_NETAPP2</zone-name>
                <zone-name>68165_NETAPP3</zone-name>
                <zone-name>lsan_68163_NETAPP</zone-name>
            </member-zone>
        </cfg>
        <zone>
            <zone-name>68161_EMC</zone-name>
            <zone-type>0</zone-type>
            <zone-type-string>zone</zone-type-string>
            <member-entry>
                <entry-name>20:00:8c:7c:ff:22:f8:80</entry-name>
                <entry-name>20:00:8c:7c:ff:22:f8:81</entry-name>
                <entry-name>50:00:09:73:58:0d:d8:5c</entry-name>
                <entry-name>50:00:09:73:58:0d:db:ff</entry-name>
            </member-entry>
        </zone>
 ...
        <zone>
            <zone-name>testzone445</zone-name>
            <zone-type>1</zone-type>
            <zone-type-string>user-created-peer-zone</zone-type-string>
            <member-entry>
                <principal-entry-name>10:00:00:00:18:00:10:20</principal-entry-name>
                <principal-entry-name>10:00:00:00:19:00:10:20</principal-entry-name>
                <principal-entry-name>10:00:00:00:23:00:10:20</principal-entry-name>
                <entry-name>10:00:00:00:15:00:10:20</entry-name>
                <entry-name>10:00:00:00:16:00:10:20</entry-name>
                <entry-name>10:00:00:00:17:00:10:20</entry-name>
                <entry-name>10:00:00:00:c9:f8:04:35</entry-name>
            </member-entry>
        </zone>
        <alias>
            <alias-name>ali30</alias-name>
            <member-entry>
                <alias-entry-name>10:00:00:00:00:00:10:01</alias-entry-name>
            </member-entry>
        </alias>
 ...
        <alias>
            <alias-name>robertTest-11-25</alias-name>
            <member-entry>
                <alias-entry-name>10:00:00:00:00:00:00:01</alias-entry-name>
            </member-entry>
        </alias>
    </defined-configuration>
 </Response>
```

DELETE Request Response

When the DELETE request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

POST Request Response

When the POST request is successful, the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

brocade-zone/effective-configuration

Description

The effective zoning configuration.

Parameters

Table 178: brocade-zone/effective-configuration

Name	Type	Description
enabled-zone [zone-name]	List	The enabled zones.
- zone-name	zoning-name-type	The name of the zone.
- zone-type	zone-type-type	The zone type. Deprecated: Although this field is deprecated, it is still returned in responses and contains valid data comparable to the 'zone-type-string' leaf.
- zone-type-string	zone-type-string-type	The zone type.
- member-entry	Container	The zone members.
-- entry-name	union {domain-index-type;fibrechannel:wwn-type;}	A list of the members in the zone. The members can be identified only as either a WWN or domain,index. Zone aliases are expanded when the configuration is enabled and are not included in the effective configuration.
-- principal-entry-name	union {fibrechannel:wwn-type;domain-index-type;}	If the zone is a peer zone, this property contains the list of principals in the zone. The members can be identified only as either a WWN or domain,index. Zone aliases are expanded when the configuration is enabled and are not included in the effective configuration.
cfg-name	zoning-name-type	The name of the effective configuration to be enabled or a configuration that is currently enabled. This must be the name of a defined configuration that is already defined. This leaf is not be present on a GET request if there is no enabled zone configuration.

Name	Type	Description
checksum	string	The checksum of the configuration database. The checksum is a MD5 (Message-Digest algorithm 5) calculation (128-bit cryptographic hash) used to verify data integrity. This resource is used by the client to verify the zone database has not changed since the configuration was read. It should be read as part of the initial query before modifications are made. The changes are accepted if the checksum agrees and rejected if it does not.
cfg-action	uint8	The action to be done to the pending zoning configuration changes. The possible values are: Read is not applicable : 0 Save the pending changes : 1 Disable the effective configuration : 2 Clear the entire zone database : 3 Clear any pending changes : 4 This value is not present with a GET request. Note that this leaf is now deprecated. Please use the cfg-action-v2 leaf instead.
cfg-action-v2	cfg-action-type	The action to be performed against the zoning configuration. See the description of cfg-action-type for possible actions to be performed.
default-zone-access	uint8	Enables or disables default zone access. The default zoning mode controls device access if zoning is not implemented or if there is no enabled zone configuration. Default zoning has two access modes: 0 : Devices in the fabric cannot access any other device in the fabric (no access). 1 : All devices within the fabric can communicate with all other devices (all access). Note that this leaf is now deprecated. Please use the default-zone-access-v2 leaf instead.
default-zone-access-v2	boolean	Enables or disables default zone access. The default zoning mode controls device access if zoning is not implemented or if there is no enabled zone configuration. Default zoning has two access modes: false : Devices in the fabric cannot access any other device in the fabric (no access). true : All devices within the fabric can communicate with all other devices (all access).
db-max	uint32	The Zone DB maximum size. The Zone DB maximum size is the upper limit for the Zone defined configuration, determined by the amount of nonvolatile memory available for storing the defined configuration. The Zone DB maximum size is further reduced due to a message header that is propagated with the zone configuration to all switches in the fabric. The maximum supported zone database size is 4 MB.
db-avail	uint32	The amount of free space in bytes available in the zone database.

Name	Type	Description
db-committed	uint32	The size in bytes of the defined configuration currently stored in nonvolatile memory. If the context is for a virtual fabric, then this resource is the size of the defined configuration for the logical switch.
db-transaction	uint32	The amount of memory required to commit the current transaction. The DB transaction size is not be zero if the defined configuration is being modified by RESTConf, Telnet, API, or other applications; otherwise, it is 0.
transaction-token	uint32	The token for the current zoning transaction on the local switch. If there is no outstanding local zoning transaction, this resource is 0. This field does not return transaction information for remote switches.
db-chassis-wide-committed	uint32	If Virtual Fabrics is in use, this is the size in bytes for the zone database across all logical switches in the chassis. If Virtual Fabrics is not in use, this is the size of the zone database for the switch. Note that the db-committed value will be smaller than this resource due to storage used for overhead.
db-chassis-wide-max	uint32	This is the maximum size in bytes available for the zone database across all logical switches in the chassis.
db-fabric-wide-max	uint32	This is the maximum size in bytes available for the zone database across the fabric. Note that this value can be different and lower than the maximum size supported by the local switch as indicated by the db-max field.
domain-with-lowest-db-max	fibrechannel:domain-id-type	This is the domain in the fabric with the lowest zone database capacity (i.e. 'db-max' value). If all switches support the same maximum size, then this field will be set to the local domain. If the local domain id is invalid or not assigned at the time of the request, then this field will be zero.

Produces

- application/xml
- application/json

Supported Methods

GET, HEAD, PATCH

Example Response

GET Request Response

When the GET operation is successful, the response has a message body similar to the following output and a “200 OK” status in the headers.

```
{
    "Response": {
        "effective-configuration": {
            "default-zone-access": 1,
            "default-zone-access-v2": true,
```

```
            "db-max": 4194304,
            "db-avail": 1036360,
            "db-committed": 5656,
            "db-transaction": 0,
            "db-chassis-wide-committed": 21715,
            "db-chassis-wide-max": 16777216,
            "db-fabric-wide-max": 1045274,
            "domain-with-lowest-db-max": 92,
            "transaction-token": 0,
            "checksum": "62c211e568f0a264ec7ab4692eac7b65",
            "cfg-name": "default",
            "enabled-zone": [
                {
                    "zone-name": "68161_EMC",
                    "zone-type": 0,
                    "zone-type-string": "zone",
                    "member-entry": {
                        "entry-name": [
                            "20:00:8c:7c:ff:22:f8:80",
                            "20:00:8c:7c:ff:22:f8:81",
                            "50:00:09:73:58:0d:d8:5c",
                            "50:00:09:73:58:0d:db:ff"
                        ]
                    }
                },
                {
                    "zone-name": "68163_EMC",
                    "zone-type": 0,
                    "zone-type-string": "zone",
                    "member-entry": {
                        "entry-name": [
                            "10:00:00:00:c9:f8:04:35",
                            "10:00:00:00:c9:f8:04:36",
                            "50:00:09:73:58:0d:d8:1c",
                            "50:00:09:73:58:0d:d8:5c"
                        ]
                    }
                },
...
                {
                    "zone-name": "68166_NETAPP3",
                    "zone-type": 0,
                    "zone-type-string": "zone",
                    "member-entry": {
                        "entry-name": [
                            "10:00:00:10:9b:8f:2d:0f",
                            "10:00:00:10:9b:8f:2d:10",
                            "20:83:00:a0:98:6b:08:7c",
                            "20:84:00:a0:98:6b:08:7c",
                            "20:85:00:a0:98:6b:08:7c",
                            "20:86:00:a0:98:6b:08:7c"
                        ]
                    }
```

```
                },
                {
                    "zone-name": "lsan_68163_NETAPP",
                    "zone-type": 0,
                    "zone-type-string": "zone",
                    "member-entry": {
                        "entry-name": [
                            "10:00:00:00:c9:f8:04:35",
                            "10:00:00:00:c9:f8:04:36",
                            "20:77:00:a0:98:6b:08:7c",
                            "20:78:00:a0:98:6b:08:7c",
                            "20:79:00:a0:98:6b:08:7c",
                            "20:7a:00:a0:98:6b:08:7c"
                        ]
                    }
                }
            ]
        }
    }
}
```


DELETE Request Response

When the DELETE request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

PATCH Request Response

When the PATCH request is successfully completed, the response contains an empty message body and a “204 No Content” status in the first line of the headers.

POST Request Response

When the POST request is successful, the resource is successfully created, the response contains an empty message body and a “201 Created” status in the first line of the headers.

REST RPC Operations

This section details the available Remote Procedure Call (RPC) operations in the FOS REST API. This section also provides information about the parameters that are associated with the RPC operations.

abort

Description

Abort RPC operation based on the input message ID.

Input

Table 179: abort

Name	Type	Description
abort-parameters	Container	The abort operation input container.
- message-id	operation:message-id-type	The input message ID which is required to abort the associated operation.

Produces

- application/xml
- application/json

configdownload

Description

The RPC command definition for the configuration download operation

on a Brocade switch.

Input

Table 180: configdownload

Name	Type	Description
configdownload-parameters	Container	The configuration download operation input parameters.
config-upload-download-option	enumeration	The supported configuration upload or download options.
port-to-area	boolean	The port-to-area addressing mode configuration files. This leaf is optional and it can be used only for the port-to-area related configdownload or upload operations.

Name	Type	Description
vf-id	fibrechannel:fabric-id-type	The fabric ID for the switch configuration to download from a switch. This leaf is valid only when you select fabric-id as the configuration upload option.
- source-vf-id	fibrechannel:fabric-id-type	The source fabric ID. The configuration from the source vf-id to be downloaded to the provided vf-id.
- config-input-buffer	string	Send the entire configuration file content as an input. The configuration payload must be encoded to base64 format. Refer to RFC 3414 for more details about the base64 encode method.

Output

Table 181: configdownload

Name	Type	Description
show-status	Container	The RPC operation status based on the specified message-id.
- message-id	message-id-type	The random message-id required to retrieve the associated operation status.
- operation	string	The operation name (such as supportsave). The show-status is common across all the operations supported in the yang RPC feature. It identifies the exact operation status when the application queries the status based on the message-id.
- status	string	The exact state of the REST operation (supportSave, firmware download, and configuration download). When an operation is triggered, the 'queued' status displays. While the operation is running the 'in-progress' status displays until all input operations are complete in the backend modules. When the input operation is complete, the 'done' status displays. After the completion status, the 'delivered' status displays until it is purged from the queue. If any operational-level errors or application-level errors occur, the system returns the standard error structure. When the input operation is removed from the queue, the 'purge' status displays.
- application-name	string	The name of the application which is used to trigger the operation. This is parsed from the HTTP USER_AGENT standard header.
- percentage-complete	uint8	The operation completion in a percentage.
- firmwaredownload	Container	A list of firmware download status messages or the EULA.
-- message	string	The status messages, depicting all stages of the firmware download progress.
-- eula-text	string	The contents of the latest EULA (Brocade End User License Agreement).
- configdownload	Container	A list of configuration download status messages or the EULA.
-- message	string	The status messages of a configuration download.

Produces

- application/xml
- application/json

Supported Methods

POST

configupload

Description

The RPC command definition for the configuration upload operation

on a Brocade switch.

Input

Table 182: configupload

Name	Type	Description
configupload-parameters	Container	The configuration upload operation input parameters.
config-upload-download-option	enumeration	The supported configuration upload or download options.
port-to-area	boolean	The port-to-area addressing mode configuration files. This leaf is optional and it can be used only for the port-to-area related configdownload or upload operations.
vf-id	fibrechannel:fabric-id-type	The fabric ID for the switch configuration to download from a switch. This leaf is valid only when you select fabric-id as the configuration upload option.

Output

Table 183: configupload

Name	Type	Description
configupload-operation-status	Container	The status of the configration upload operation.
- status-message	string	The status message. This is an information message about the configuration upload operation.
- config-output-buffer	string	Retrieves the entire configuration for the specified configuration upload option. The output is encoded to base 64 format. Refer to RFC 3414 for more details about the base64 encode method.

Produces

- application/xml
- application/json

Supported Methods

POST

date

Description

The container for date commands.

Input

Table 184: date

Name	Type	Description
date-operation-parameters	Container	The container to configure the system date.
- day-of-month	uint16	The day of the month.
- month	brocade-operation-types:month-type	The month of the year. (for example, jan, feb, mar, apr, and so on).
- year	uint16	The year. The system supports the year from 1970.
- hour	uint16	The hour.
- minute	uint16	The minute.

Output

Table 185: date

Name	Type	Description
date-operation-status	Container	The status of the date operation.
- status-message	string	This is status message to show the errors or success of date operation.

Produces

- application/xml
- application/json

Supported Methods

POST

device-login-rebalance

Description

The RPC command definition to manage the ports connected

to Fabric Interconnect (FI).

Input

Table 186: device-login-rebalance

Name	Type	Description
device-login-rebalance-parameters	Container	The device login actions container.
- node-wwn	fibrechannel:wwn-type	The node WWN of the Fabric Interconnect (FI).

Produces

- application/xml
- application/json

device-management

Description

The RPC command definition for Brocade device management operations.

Input

Table 187: device-management

Name	Type	Description
device-management-group		The input parameters used for the device management operations.

Output

Table 188: device-management

Name	Type	Description
device-management-group		The device response container.

Produces

- application/xml
- application/json

Supported Methods

POST

extension

Description

The RPC command definition for Extension operation on a Brocade

switch.

Input

Table 189: extension

Name	Type	Description
extension-operation-parameters	Container	The extension operation input container.
- slot	uint16	The slot number for the data path processor.
- config-clear	boolean	Clears the extension configuration on the extension switch or blade. Note that the config-clear operation may be disruptive. For extension switches, you must reboot the switch after the config-clear operation; otherwise, configuration operations are blocked. For extension blades, you must power off the blade before you clear the slot's corresponding configuration. Supported values: false: No operation. true : Clears the extension configuration.
- config-default	boolean	Resets the default configuration on the extension switch or blade. Note that the config-default operation is disruptive. For extension switches, the extension switch reboots automatically after a successful config-default operation. For extension blades, make sure the blade is in an online state for the default-config operation. You may need to power off and on the blade slot to apply different extension modes correctly. Supported values: false: No operation. true : Resets the default configuration on the extension device.
- global-lan-statistics-reset	boolean	Resets the global LAN statistics on a data processor (DP). Supported values: false: No operation. true : Resets the global LAN statistics on a DP.
- dp-id	extension:dp-id-type	The extension data path processor ID. Depending on the extension platform, the device may have a single DP or dual DP. A single DP supports DP0 only. A dual DP supports both DP0 and DP1. 0 : DP0 1 : DP1.

Output

Table 190: extension

Name	Type	Description
extension-operation-status	Container	The container for the status of the extension operation.
- status-message	string	The status message (success or fail) for the operation.

Produces

- application/xml
- application/json

Supported Methods

POST

fibrechannel-fabric

Description

The RPC command definition for fabric operation.

Input

Table 191: fibrechannel-fabric

Name	Type	Description
fabric-operation-parameters	Container	Input container to perform fabric related operation.
- build-fabric	boolean	The input which is required to initiate the build fabric operation. This operation requires switch to be enabled state. And only 'true' should be used as input. To verify the build fabric status please infer brocade-fibrechannel-fabric?

Produces

- application/xml
- application/json

Supported Methods

POST

factory-reset

Description

The RPC command definition for the factory reset operation on a Brocade

switch.

Input

Table 192: factory-reset

Name	Type	Description
reset-option-parameters	Container	Performs a factory reset operation on a specified feature.
- clear-features	Container	A list of configurations that you want to clear during the factory reset operation. To clear all configurations, specify all.
-- feature	union {brocade-operation-types:clear-feature-type;string;}	The feature configuration that you want to clear during the factory reset operation.
- keep-features	Container	A list of configurations that you want to keep during the factory reset operation.
-- feature	brocade-operation-types:keep-feature-type	The feature configuration that you want to keep during the factory reset operation.
- set-features	Container	A list of configurations that you want to set during the factory reset operation.
-- feature	brocade-operation-types:set-feature-type	The feature configuration that you want to set during the factory reset operation.

Output

Table 193: factory-reset

Name	Type	Description
factory-reset-operation-status	Container	The factory reset operation status.
- status-message	string	The status message (success or failure) of the factory reset operation. The operation status is returned only if the keep feature IP is set or set feature is provided; otherwise, a successful factory reset resets the IP and the operation status times out.

Produces

- application/xml
- application/json

Supported Methods

POST

firmwaredownload

Description

The RPC command definition to trigger the firmware download operation.

Input

Table 194: firmwaredownload

Name	Type	Description
firmwaredownload-parameters	Container	The firmware download operation input container.
- eula-action	operation-types:eula-action-type	The End User License Agreement(EULA) actions for firmware download. Note that the firmware download process requires that you accept the EULA. .
- host	inet:host	The IPv4 or IPv6 address or host name for the remote server.
- user-name	string	The user name of the remote server that is used to copy the firmware download files.
- password	string	The password for the remote server used to copy the firmware download files. The password should be base64 encoded. Refer to RFC 3414.
- remote-directory	string	The fully qualified path to the remote server directory from which to transfer the firmware download.
- protocol	operation-types:protocol	The file transfer protocol. The values are ftp (File Transfer Protocol), scp (Secure Copy Protocol), sftp (Secure FTP), or http (HyperText Transfer Protocol).
- port	uint32	The user-defined port number for the specified file transfer protocol (scp, sftp, or http).
- stage	boolean	Downloads the firmware to a secondary partition without rebooting and activation.
- activate	boolean	Activates the firmware, if not already activated. true: Activates the firmware. Do not include any other input parameters with this action.
- firmware-sync	boolean	Synchronizes the firmware on the active CP to the standby CP. true: Synchronizes the firmware. Do not include any other input parameters with this action.
- clean-install	boolean	Initiates a clean reinstall of the firmware.

Output

Table 195: firmwaredownload

Name	Type	Description
show-status	Container	The RPC operation status based on the specified message-id.
- message-id	message-id-type	The random message-id required to retrieve the associated operation status.
- operation	string	The operation name (such as supportsave). The show-status is common across all the operations supported in the yang RPC feature. It identifies the exact operation status when the application queries the status based on the message-id.
- status	string	The exact state of the REST operation (supportSave, firmware download, and configuration download). When an operation is triggered, the 'queued' status displays. While the operation is running the 'in-progress' status displays until all input operations are complete in the backend modules. When the input operation is complete, the 'done' status displays. After the completion status, the 'delivered' status displays until it is purged from the queue. If any operational-level errors or application-level errors occur, the system returns the standard error structure. When the input operation is removed from the queue, the 'purge' status displays.
- application-name	string	The name of the application which is used to trigger the operation. This is parsed from the HTTP USER_AGENT standard header.
- percentage-complete	uint8	The operation completion in a percentage.
- firmwaredownload	Container	A list of firmware download status messages or the EULA.
-- message	string	The status messages, depicting all stages of the firmware download progress.
-- eula-text	string	The contents of the latest EULA (Brocade End User License Agreement).
- configdownload	Container	A list of configuration download status messages or the EULA.
-- message	string	The status messages of a configuration download.

Produces

- application/xml
- application/json

Supported Methods

POST

license

Description

The RPC command definition for license operation on a Brocade

switch.

Input

Table 196: license

Name	Type	Description
license-parameters	Container	The license operation input container.
- name	string	The representation of the license would be either the license key or serial number. The license key is a string with alpha numeric characters and the License serial number is a string with the format of 'FOS-XX-X-XX-XXXXXXXX'. An example of a license key and serial number are as follows: License key: 'HP9ttZNSgmB4MCD3NmNWgQDWtAKBFtXtBSFJF' Serial number: 'FOS-00-0-02-11201234' This leaf is not included when installing the licenses with serial number from a file. Use this leaf to export or remove a specific license certificate.
- action	enumeration	Action against a specified license
- license-payload	string	Use to send the entire license certificate content as a input. The license certificate payload must be base64 encoded to avoid the nested xml tag issue during input. Refer to RFC 3414 for more details about base64 encode method.
- usb-file-path	string	The relative path of the license certificate present in a USB. This leaf is required only when the user wanted to install the license certificate from USB device.
host	inet:host	The IP address or host name of the remote server.
user-name	string	The user name of the remote server that is used to copy the license certificates.
password	string	The password for the remote server used to copy the license certificate. The password must be base64 encoded. Refer to RFC 3414.
remote-directory	string	The file path of the remote server from which the license certificate is to be transferred.
protocol	operation-types:protocol	The transport protocol.
port	uint32	The user-defined port number for scp (Secure Copy Protocol) and sftp (Secure FTP).
remote-file	string	The remote file name is used to export the license certificate. This leaf is required only when you want to export the license certificate into a specified remote filename.

Output

Table 197: license

Name	Type	Description
license-operation-status	Container	The status of the license operation.
- status-message	string	The status message. This is an informational message for you to perform additional actions such as reboot, port/switch/chassis disable or enable for the license to take effect.

Produces

- application/xml
- application/json

Supported Methods

POST

lldp

Description

The RPC command definition for LLDP port operations

on a Brocade switch.

Input

Table 198: lldp

Name	Type	Description
lldp-operations	Container	The LLDP actions container.
- slot-port	fibrechannel:slot-port-name-type	The port number in slot/port format.
- cp-name	fibrechannel:cp-name-type	For a fixed-port switch, the valid name is cp0. For a chassis switch, the valid names are cp0 and cp1.
- physical-interface	fibrechannel:physical-interface-type	The physical management iterface name. For a fixed-port switch, the valid name is eth0. For a chassis switch, the valid names are eth0 and eth3.
- action	enumeration	The LLDP actions on the specified port.

Produces

- application/xml
- application/json

Supported Methods

POST

login

Description

The RPC command definition to get the login response.

Output

Table 199: login

Name	Type	Description
switch-parameters	Container	Contains switch and user details. Upon login success, application can use them to initialize the session.
- user-name	brocade-security-type:user-config-login-user-name-type	The name of the account used at login.
- user-role	brocade-security-type:user-config-role-type	The user role for the switch in non-VF mode.
- chassis-access-role	brocade-security-type:user-config-role-type	Specifies the account's access permissions regarding chassis-level commands.
- home-virtual-fabric	brocade-security-type:home-virtual-fabric-type	Specifies the account's home Virtual Fabric. If the value is not configured in ldap-role-map, it will return '0' value.
- virtual-fabric-enabled	boolean	Whether Virtual Fabrics is enabled on the chassis. true : Virtual Fabrics is enabled on the chassis. false : Virtual Fabrics is not enabled on the chassis.
- virtual-fabric-supported	boolean	Whether Virtual Fabrics is supported on the chassis. true : Virtual Fabrics is supported on the chassis. false : Virtual Fabrics is not supported on the chassis.
- days-to-user-password-expiration	uint32	The time before expiration for the user to receive a warning message. Response does not include this leaf, if logged in user's password never expires.
- firmware-version	string	A human readable string identifying the active firmware version running on the switch.
- model	string	The model name of the switch.
- virtual-fabric-role-ids	Container	The Virtual Fabrics roles and IDs assigned to the user account.
-- role-id	brocade-security-type:virtual-fabric-role-id-type	The Virtual Fabrics assigned to the user account.
- virtual-fabric-ids	Container	The virtual fabric list available in the switch for the logged in user.
-- fabric-id	fibrechannel:fabric-id-type	The virtual fabric identification (VFID) of the logical switch.

Produces

- application/xml
- application/json

management-ethernet-interface

Description

The RPC command definition for operations on the management

Ethernet interface of switch.

Input

Table 200: management-ethernet-interface

Name	Type	Description
ethernet-interface-parameters	Container	The input container for the management Ethernet interface operation.
- physical-interface	fibrechannel:physical-interface-type	The name of the physical management Ethernet interface on which to perform the operation. For a fixed-port switch, the valid value is eth0. For a chassis switch, the valid values are eth0 and eth3.
- cp-name	fibrechannel:cp-name-type	The host CP name that identifies the CP on which to perform the action on its management Ethernet interface.
- action	enumeration	The action (such as, reset-error) to perform on the management Ethernet interface.

Produces

- application/xml
- application/json

Supported Methods

POST

maps-policy-restore

Description

The RPC command definition to restore the default policy.

Input

Table 201: maps-policy-restore

Name	Type	Description
maps-policy-restore-parameters	Container	Indicates the maps policy restore parameters - one, or multiple policies.
- name	maps-types:maps-rule-name-type	The default rule name to be restored. if rule name os empty then all the modified rules will be restored.

Produces

- application/xml
- application/json

maps-rule-restore

Description

The RPC command definition to restore the default rule.

Input

Table 202: maps-rule-restore

Name	Type	Description
maps-rule-restore-parameters	Container	Indicates the default maps rule restore parameters - one, or multiple rules.
- name	maps-types:maps-rule-name-type	The default rule name to be restored. if rule name os empty then all the modified rules will be restored.

Produces

- application/xml
- application/json

ntp-clock-server

Description

The RPC command definition for Brocade time server operations.

Input

Table 203: ntp-clock-server

Name	Type	Description
clock-server-parameters	List	Configures clock server and server key index.
- action	brocade-time-type:ts-operation-type	The action to configure clock server list and server key index list. Available actions include: o create o modify

Produces

- application/xml
- application/json

pcie-health-test

Description

The RPC command definition for Brocade PCIe health operations.

Input

Table 204: pcie-health-test

Name	Type	Description
slot-test	Container	The input parameters for the PCIe health slot test query.
- slot-id	uint8	The physical slot number in the chassis where the blade is inserted. A value of 255 specifies all slots in the chassis.

Output

Table 205: pcie-health-test

Name	Type	Description
slot-test	List	The response parameters for the PCIe slot health test.
- slot-id	uint8	The physical slot number in the chassis where the blade is inserted.
- status	string	The status (sucess or failure) of testing the PCIe links between the non-transparent ports of the PCIe switch in the blades and the standby CP.

Produces

- application/xml
- application/json

Supported Methods

POST

port

Description

The RPC command definition for port operations.

Input

Table 206: port

Name	Type	Description
port-operation-parameters	List	The parameters required for port operations.
- name	fibrechannel:slot-port-name-type	The port name in slot/port.
- operation-type	port-operation-type	The operation being performed for the port.
- user-port-address	fibrechannel:area-address-type	Port address to be bound/unbound in hex format.
- auto-bind	boolean	true: Enables autobinding on the specified port. If the auto feature is enabled, the entire area field of the N_Port ID is bound to a single port. With 10-bit routing, up to 4 ports can share the same 8-bit area field of the N_Port ID. This address assignment mode dedicates all four unique routes to a single port. false: Auto-binding disabled

Produces

- application/xml
- application/json

Supported Methods

POST

port-decommission

Description

The RPC command for triggering the port decommissioning operation.

Input

Table 207: port-decommission

Name	Type	Description
port-decommission-parameters	List	The parameters required for port operations. .
- slot-port	fibrechannel:slot-port-name-type	The slot/port of the port.
- port-decommission-type	port-decommission-operation-type	The operation being performed for the port.

Output

Table 208: port-decommission

Name	Type	Description
show-status	Container	The RPC operation status based on the specified message-id.
- message-id	message-id-type	The random message-id required to retrieve the associated operation status.
- operation	string	The operation name (such as supportsave). The show-status is common across all the operations supported in the yang RPC feature. It identifies the exact operation status when the application queries the status based on the message-id.
- status	string	The exact state of the REST operation (supportSave, firmware download, and configuration download). When an operation is triggered, the 'queued' status displays. While the operation is running the 'in-progress' status displays until all input operations are complete in the backend modules. When the input operation is complete, the 'done' status displays. After the completion status, the 'delivered' status displays until it is purged from the queue. If any operational-level errors or application-level errors occur, the system returns the standard error structure. When the input operation is removed from the queue, the 'purge' status displays.
- application-name	string	The name of the application which is used to trigger the operation. This is parsed from the HTTP USER_AGENT standard header.
- percentage-complete	uint8	The operation completion in a percentage.
- firmwaredownload	Container	A list of firmware download status messages or the EULA.
-- message	string	The status messages, depicting all stages of the firmware download progress.
-- eula-text	string	The contents of the latest EULA (Brocade End User License Agreement).
- configdownload	Container	A list of configuration download status messages or the EULA.
-- message	string	The status messages of a configuration download.

Produces

- application/xml
- application/json

Supported Methods

POST

portchannel

Description

The RPC command definition for portchannel operations

on a Brocade switch.

Input

Table 209: portchannel

Name	Type	Description
portchannel-operation-parameters	Container	portchannel actions container
- name	brocade-interface-types:portchannel-interface-name-type	The portchannel interface name.
- action	enumeration	portchannel actions on the given portchannel

Produces

- application/xml
- application/json

Supported Methods

POST

reboot

Description

The RPC command definition for different reboot operations on a Brocade

switch.

Input

Table 210: reboot

Name	Type	Description
reboot-operation-parameters	Container	The option to perform different reboot operations on a specified switch.
- action	brocade-operation-types:reboot-types	Specify particular way to reboot the switch.

Output

Table 211: reboot

Name	Type	Description
reboot-operation-status	Container	reboot operation status.
- status-message	string	This is status message to user for the errors or success of reboot operation.

Produces

- application/xml
- application/json

Supported Methods

POST

restart

Description

The RPC command definition for restart operations.

Input

Table 212: restart

Name	Type	Description
restart-operation-parameters	List	The rpc parameters required for restart operations.
- action	restart-operation-action-type	The restart action operation to be performed. Actions: restart-ike-sessions: The action 'restart-ike-sessions' can be used to restart the IPsec policy IKE sessions. and deprecates the leaf 'restart-ike-sessions' from 'brocade-extension-ipsec-policy' for 'extension-ipsec-policy' list.
- extension-ipsec-policy-name	extension:extension-name-type	The IPsec policy name for restart of IKE sessions.

Produces

- application/xml
- application/json

sdd-quarantine

Description

The RPC command definition for operations on slow drain device (sdd) quarantined ports.

Input

Table 213: sdd-quarantine

Name	Type	Description
sdd-quarantine-parameters	Container	The slow drain device quarantine container.
- action	enumeration	Action to be taken for local switch quarantined ports
- port-index	fibrechannel:port-number-type	The port index of the local switch quarantined port. Value is same as that of leaf port-index of the list quarantined-devices.
- force	boolean	Allows clearing of quarantined state from the local switch ports even when the latency or frame loss condition persists.

Produces

- application/xml
- application/json

security-acl-policy

Description

The RPC command definition for security policy operation on a Brocade switch.

Input

Table 214: security-acl-policy

Name	Type	Description
security-policy-parameters	List	List of policy parameters for configuring SCC/FCS/DCC.
- policy-name	brocade-security-type:security-policy-name-type	The security policy name. Valid Values are - SCC_POLICY - FCS_POLICY - DCC_POLICY_* Policy name name must be capitalized. The policy name DCC_POLICY_* can be up to 30 characters, including the prefix DCC_POLICY_ ALL_DCC_POLICY, ALL_STALE_DCC_POLICY are valid for delete instance.

Name	Type	Description
- include-device	boolean	Specifies whether device(s) connected to the port should be included in the policy or not. true : Connected device(s) will be included. false: Connected device(s) will NOT be included.
- action	brocade-security-type:security-policy-action-type	The action to take on the security policy. Security policy configurations are saved automatically after any change.
- fcs-position	uint16	Specifies the to position of the FCS switch for FCS move.
- members	Container	List of policy members for SCC/FCS/DCC policy.
-- member	brocade-security-type:security-input-type	Information for configuring security policy. For DCC_POLICY, only one member is allowed per DCC policy object.
- switch-ports	Container	Port list
-- port	union {fibrechannel:user-port-number-type;string;}	Port can be specified by asterisk (*) or by port index. Asterisk (*) specifies all ports in the switch.

Produces

- application/xml
- application/json

Supported Methods

POST

security-authentication-configuration

Description

The RPC command definition for modifying authentication settings.

Equivalent to the CLI authutil.

Input

Table 215: security-authentication-configuration

Name	Type	Description
authentication-configuration-parameters	Container	Performs the authentication operations.
- action	brocade-security-type:authentication-operation-type	Operations pertaining to authentication

Name	Type	Description
- authentication-init-slot-ports	Container	Reinitiates authentication on selected ports after changing the DH-CHAP group, hash type and shared secret between a pair of switches. This operation does not work on private, loop, NPIV, and FICON devices. This operation may bring down the E_Ports if the DH-CHAP shared secrets are not installed correctly. This operation is not supported on encrypted ports. This option is not supported in AG mode. This operation does not initiate authentication on disabled ports.
-- port	brocade-security-type:authentication-initiation-parameter-type	Specifies the slot port for initiating authentication.

Produces

- application/xml
- application/json

Supported Methods

POST

security-authentication-secret

Description

The RPC command definition for configuring shared secrets for DH-CHAP authentication.

Input

Table 216: security-authentication-secret

Name	Type	Description
authentication-secret-parameters	List	Manages the DH-CHAP shared secret key database used for authentication.
- member	brocade-security-type:security-input-type	Switch or node for which shared secret to be configured. Values can be switch name or switch WWN or domain ID or node WWN. In Access Gateway mode, only a WWN is allowed. Wildcard (*) is allowed only for remove. Use this value to remove all secret pair entries in the switch.
- peer-secret	brocade-security-type:authentication-secret-type	The secret of the peer that authenticates peer. The secret must be encoded with the Base64 encoding scheme.
- local-secret	brocade-security-type:authentication-secret-type	The local secret that authenticates peer. The local and peer secret should not be the same. The secret must be encoded with the Base64 encoding scheme.

Name	Type	Description
- action	brocade-security-type:authentication-secret-action-type	The action to Manage the DH-CHAP shared secret key information. Available actions include: o set o remove

Produces

- application/xml
- application/json

Supported Methods

POST

security-certificate

Description

The RPC command definition for Brocade security certificate operations.

Input

Table 217: security-certificate

Name	Type	Description
security-certificate-parameters	Container	The use of third-party certificates on a switch, including Public Key Infrastructure (PKI) based certificates, Lightweight Directory Access Protocol (LDAP) certificates, RADIUS certificates, syslog CA certificates and extension certificates.
- certificate-entity	brocade-security-type:certificate-entity-type	The certificate entity (Switch, CA, CA Client, or Server certificate).
- certificate-type	brocade-security-type:certificate-application-type	The certificate type (commoncert, https, radius, ldap, syslog, fcap or extension).
- certificate-name	string	The certificate name.
- ca-certificate	string	The preimported local CA certificate name.
- keypair-tag	string	The keypair tag is a user-defined name used to identify a local certificate uniquely.
- action	brocade-security-type:seccertmgmt-action-type	The security certificate operations (such as import).
- certificate-hexdump	brocade-security-type:certificate-hexdump-type	The CA-client, CA-server, or switch certificate contents in raw hexadecimal format, including beginning and end markers, used on the switch.

Produces

- application/xml
- application/json

Supported Methods

POST

security-fabric-wide-policy-distribute

Description

The RPC command definition for setting the fabric-wide consistency policy.

Input

Table 218: security-fabric-wide-policy-distribute

Name	Type	Description
distribute-parameters	Container	Parameters for setting the fabric-wide consistency policy.
- policy-types	Container	Security database(s) to be distributed.
-- type	brocade-security-type:fabric-wide-consistency-policy-type	All members specified in a given policy set are automatically distributed to all participating switches in the fabric that support the policy.

Produces

- application/xml
- application/json

Supported Methods

POST

security-policy-chassis-distribute

Description

The RPC command definition for chassiss wide distribution of IP filter polices on Brocade switch(es).

Input

Table 219: security-policy-chassis-distribute

Name	Type	Description
policy-chassis-distribute-parameters	Container	Parameters for chassis distribution of security database.
- policy-types	Container	Security database(s) to be distributed.
-- type	brocade-security-type:chassis-distribute-policy-type	Policy name of the security database(s) to be distributed.

Name	Type	Description
- destination-domains	Container	List of switches that should receive the data distribution.
-- domain-id	brocade-security-type:chassis-distribute-domain-type	Domain ID that should receive the data distribution. A wildcard (*) may be specified to include all switches in the fabric. This operand is required and valid only when Virtual Fabrics are disabled.
- vf-id	fibrechannel:fabric-id-type	Specifies the fabric ID that should receive IP filter distribution. The valid values for FID are from 1 through 128. Database is distributed to all chassis that are connected to the specified logical switch (FID) and are configured to accept the distribution. This operand is required and valid only when the originating switch is in Virtual Fabric mode.

Output

Table 220: security-policy-chassis-distribute

Name	Type	Description
chassis-distribute-operation-status	List	Chassis distribute operation status
- vf-id	fibrechannel:fabric-id-type	Specifies the fabric ID that received IP filter distribution. The valid values for FID are from 1 through 128. Database is distributed to all chassis that are connected to the specified logical switch (FID) and are configured to accept the distribution. This field will be empty for NON-VF destination.
- distribute-supported	boolean	Specifies whether the the distribution is supported on the receiving chassis. true : Specified chassis supports distribution. false: Specified chassis does not support distribution.

Produces

- application/xml
- application/json

Supported Methods

POST

security-policy-distribute

Description

The RPC command definition for security policy distribution on Brocade switch(es).

Input

Table 221: security-policy-distribute

Name	Type	Description
policy-distribute-parameters	Container	Parameters for distributing security database.
- distribute-policy-types	Container	List of policy sets, also called as security databases, to be distributed.
-- type	brocade-security-type:distribute-policy-type	Policy name of the security databases to be distributed.
- destination-names	Container	List of switches that should receive the data distribution.
-- name	brocade-security-type:security-input-type	Switch that should receive the data distribution. Values can be switch name or switch WWN or domain ID. A wildcard (*) may be specified to include all switches in the fabric.

Output

Table 222: security-policy-distribute

Name	Type	Description
distribute-operation-status	Container	distribution operation status
- status-message	string	The status message. This message is an information for the user that the Distribution operation succeeded for the mentioned domain(s).
- access-gateway-nodes	Container	When Password Policy is distributed to an AG or switch connected to an AG, the list of AG's to which the Policy is distributed are. listed. Distribution of other policies or combination of Password Policy along with other policy will not list AG information.
-- node	string	Name and wwn of the AG switch. Example: myAccessGateway@11:22:33:44:55:66:77:88

Produces

- application/xml
- application/json

Supported Methods

POST

security-reset-violation-statistics

Description

The RPC command definition for resetting security violation statistics.

Input

Table 223: security-reset-violation-statistics

Name	Type	Description
violation-statistics	Container	Parameters for resetting the security violation statistic(s).
- domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
- statistics-name	brocade-security-type:security-violation-statistics-type	Name of the security violation statistics to be reset.

Produces

- application/xml
- application/json

Supported Methods

POST

security-role-clone

Description

The RPC command definition to trigger a security role clone operation.

Input

Table 224: security-role-clone

Name	Type	Description
security-role-clone-parameters	Container	The security role clone operation input container.
- name	brocade-security-type:user-config-role-type	Specifies the new role name.
- source-name	brocade-security-type:user-config-role-type	Specifies name of the role to be cloned.

Produces

- application/xml
- application/json

show-status

Description

The RPC asynchronous operation status based on the input message ID.

Input

Table 225: show-status

Name	Type	Description
message-id	operation:message-id-type	The input message ID which is required to retrieve the associated operation status.

Output

Table 226: show-status

Name	Type	Description
show-status	Container	The RPC operation status based on the specified message-id.
- message-id	message-id-type	The random message-id required to retrieve the associated operation status.
- operation	string	The operation name (such as supportsave). The show-status is common across all the operations supported in the yang RPC feature. It identifies the exact operation status when the application queries the status based on the message-id.
- status	string	The exact state of the REST operation (supportSave, firmware download, and configuration download). When an operation is triggered, the 'queued' status displays. While the operation is running the 'in-progress' status displays until all input operations are complete in the backend modules. When the input operation is complete, the 'done' status displays. After the completion status, the 'delivered' status displays until it is purged from the queue. If any operational-level errors or application-level errors occur, the system returns the standard error structure. When the input operation is removed from the queue, the 'purge' status displays.
- application-name	string	The name of the application which is used to trigger the operation. This is parsed from the HTTP USER_AGENT standard header.
- percentage-complete	uint8	The operation completion in a percentage.
- firmwaredownload	Container	A list of firmware download status messages or the EULA.
-- message	string	The status messages, depicting all stages of the firmware download progress.
-- eula-text	string	The contents of the latest EULA (Brocade End User License Agreement).
- configdownload	Container	A list of configuration download status messages or the EULA.
-- message	string	The status messages of a configuration download.

Produces

- application/xml
- application/json

Supported Methods

POST

slot-configuration

Description

The RPC command definition to set default configration for the given slot.

Input

Table 227: slot-configuration

Name	Type	Description
slot-configuration-parameters	Container	The slot configuration operation input container.
- slot-number	uint16	The number of the physical slot in the chassis in which the blade is inserted.
- action	slot-configuration-action-type	The actions to be performed on the slot.
- reset	boolean	true: Move ALL ports to Default Switch (DS) regardless of which Logical Switch the port(s) is being configured and set ALL ports of given slot number to default port configuration. We must run slotpoweroff and slotpoweron commands via CLI to complete this recovery process of slot. false: No operation as this is not supported.
- power-state	power-state-type	Indicates the slot to manually power off or power on.

Produces

- application/xml
- application/json

supportlink

Description

The RPC command definition to trigger the supportLinkProfile operation.

Input

Table 228: supportlink

Name	Type	Description
supportlink	Container	The supportLinkProfile operation input container.
- action	enumeration	The action like default, collect, or send to be performed.

Produces

- application/xml
- application/json

Supported Methods

POST

traffic-optimizer

Description

The RPC command definition for traffic optimizer operations

on a Brocade switch.

Input

Table 229: traffic-optimizer

Name	Type	Description
traffic-optimizer-parameters	Container	traffic optimizer actions container
- action	enumeration	Traffic Optimizer actions for the given profile
- name	string	Traffic Optimizer profile name
- delay	uint16	This will schedule the time to activate the profile 1 to 60 - Activate the profile after given num of days
- protocol	enumeration	The protocol Name for a profile
- speeds	Container	List of speeds supported on custom profile when Protocol is SCSI, 4-gfc, 8-gfc, 16-gfc 32-gfc, 64-gfc speeds are supported. When protocol is NVMe, 32-gfc, 64-gfc speeds are supported.
-- speed	fibrechannel:protocol-speed-type	Speed supported on custom profile

Produces

- application/xml
- application/json

Supported Methods

POST

usb-delete-file

Description

The RPC command definition for usb operation.

Input

Table 230: usb-delete-file

Name	Type	Description
usb-operation-parameters	Container	Input container to perform usb related operation.
- path	operation-types:usb-path-type	Path to the file.
- name	operation-types:usb-filename-type	Name of file in the USB device.

Produces

- application/xml
- application/json

fibrechannel-zone

Description

The RPC command definition for zone operations.

Input

Table 231: fibrechannel-zone

Name	Type	Description
zone-operation-parameters	List	Input list of parameters specifying what type of zone operations to perform.
- action	zone-operation-action-type	The type of RPC zone action operation to perform.
- zone-object	zone:zone-member-type	The zone object upon which the RPC zone action will be performed.
- new-zone-object	zone:zone-member-type	The new zone object that will be the result of the RPC zone action.

Produces

- application/xml
- application/json

Supported Methods

POST

Definitions

The FOS REST API supports additional data types (type definitions, features, and groupings). Additional details about these data types can be found in a data type file (for example, fibrechannel-yang-types.yang) or the associated .yang file (for example, brocade-zone.yang). These files are located in the same folder as the module .yang files.

aaa-authspec-type

Name	Type	Value	Description
aaa-authspec-type	string	length: 1..128 pattern: radius;local| radius;localbackup|tacacs\+;local| tacacs\+;localbackup|ldap;local| ldap;localbackup|radius|tacacs\+| ldap|local	Authentication mode for RADIUS, TACACS+ and LDAP.

aaa-encryption-algorithm-type

Name	Type	Value	Description
aaa-encryption-algorithm-type	enumeration	none: Null encryption algorithm aes256: AES-256 algorithm	Encryption algorithm type.

aaa-protocols-type

Name	Type	Value	Description
aaa-protocols-type	enumeration	chap: Challenge-Handshake Authentication Protocol pap: Password Authentication Protocol	Authentication protocols that are commonly used by both RADIUS and TACACS+.

aaa-timeout-type

Name	Type	Value	Description
aaa-timeout-type	uint16	range: 1..30	Response timeout for the RADIUS, TACACS+ and LDAP.

access-permission

Name	Type	Value	Description
access-permission	enumeration	ro: The access level is read only. rw: The access level is read write.	The SNMP access control permission.

aaa-authspec-type

Name	Type	Value	Description
aaa-authspec-type	string	length: 1..128 pattern: radius;local| radius;localbackup|tacacs\+;local| tacacs\+;localbackup|ldap;local| ldap;localbackup|radius|tacacs\+| ldap|local	Authentication mode for RADIUS, TACACS+ and LDAP.

authentication-data

Name	Type	Value	Description
authentication-data	union	preshared-key pki-key-pair fibrechannel:brocade-empty-string-type	Specifies the authentication-data depending on the profile used.

ag-mode-type

Name	Type	Value	Description
ag-mode-type	enumeration	not-supported: AG mode is not supported by this device. disabled: AG mode is supported and is currently disabled. enabled: AG mode is supported and is currently enabled.	Indicates the Access Gateway (AG) mode capability and state. A switch is capable of AG mode support when this value is not supported.

air-flow-direction-type

Name	Type	Value	Description
air-flow-direction-type	string	length: 1..255 pattern: forward (non-portside exhaust)|non-portside exhaust| reverse (non-portside intake)|non-portside intake|not available|not applicable	The air flow direction of the fan.

alarm-type

Name	Type	Value	Description
alarm-type	uint32	range: 0..max	The high or low alarm type for the media.

alert-type

Name	Type	Value	Description
alert-type	int16	range: -32767..32767	The high or low alert type for the media.

area-address-type

Name	Type	Value	Description
area-address-type	string	length: 0|6 pattern: (0[xX][0-9a-fA-F]{4})?	The 16-bit area address format - for example: 0x0A40

arl-algo-type

Name	Type	Value	Description
arl-algo-type	enumeration	auto: The 'auto' ARL to auto select either of the 'static-reset' 'step-down' or 'timed-step-down' algorithm. This is used only in configuration. static-reset: The ARL 'static-reset' algorithm resets to the minimum configured bandwidth for the circuit when congestion is detected at the current bandwidth, and restarts the climb toward maximum bandwidth. step-down: The ARL 'step-down' algorithm is used to reduce the configured bandwidth when RTT < 200ms. timed-step-down: The 'timed-step-down' ARL reduces rate when RTT >= 200ms.	The different adaptive rate limiting support.

as-number

Name	Type	Value	Description
as-number	uint32		The as-number type represents autonomous system numbers which identify an Autonomous System (AS). An AS is a set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASes. IANA maintains the AS number space and has delegated large parts to the regional registries. Autonomous system numbers were originally limited to 16 bits. BGP extensions have enlarged the autonomous system number space to 32 bits. This type therefore uses an uint32 base type without a range restriction in order to support a larger autonomous system number space. In the value set and its semantics, this type is equivalent to the InetAutonomousSystemNumber textual convention of the SMIv2.

ascii-text-type

Name	Type	Value	Description
ascii-text-type	string	pattern: [-~]*	The type of text.

authentication-hash-type

Name	Type	Value	Description
authentication-hash-type	enumeration	sha256: Specifies SHA-256 hash function. sha1: Specifies SHA-1 hash function. md5: Specifies MD5 hash function. sha1-md5: Specifies both SHA-1 and MD5 hash function. all: Specifies all the supported hash function.	Authentication hash type.

authentication-initiation-parameter-type

Name	Type	Value	Description
authentication-initiation-parameter-type	union	fibrechannel:slot-port-name-type authentication-initiation-type	Input for intitiating authentication.

authentication-initiation-type

Name	Type	Value	Description
authentication-initiation-type	enumeration	all-e-ports: Specifies all E_Ports in the switch.	Input for initiating authentication in specified type of ports.

authentication-operation-type

Name	Type	Value	Description
authentication-operation-type	enumeration	authentication-init: Reinitiates authentication on selected ports.	Modifies the authentication configuration.

authentication-policy-mode-type

Name	Type	Value	Description
authentication-policy-mode-type	enumeration	off: Turns the authentication policy off, and the switch rejects any authentication requests. passive: Sets the authentication policy to passive mode (default). Authentication is not initiated but participates in authentication, if the connecting switch/device initiates it. active: Sets the authentication policy to active mode. During switch initialization, authentication is initiated on all E_Ports, but the port is not disabled if the connecting switch does not support authentication or the authentication policy is turned off. Applicable only for switch authentication policy. on: Authentication is mandatory. If the attached device is not capable of doing authentication, the corresponding port is disabled.	Authentication policy modes.

authentication-protocol-type

Name	Type	Value	Description
authentication-protocol-type	enumeration	fcap: Specifies FCAP as only authentication protocol. dh-chap: Specifies DH-CHAP as only authentication protocol. all: Specifies both FCAP and DH-CHAP authentication protocol.	Authentication Protocol.

authentication-secret-action-type

Name	Type	Value	Description
authentication-secret-action-type	enumeration	set: Sets the shared secret pair. remove: Removes the DH-CHAP shared secret pair. Supply asterisk (*) for removing all secrets	Manages the DH-CHAP shared secret key information.

authentication-secret-type

Name	Type	Value	Description
authentication-secret-type	string	length: 12..40 pattern: [-~]+ pattern: : modifier: invert-match	Specifies a shared secrets for DH-CHAP authentication on F_Port, N_Port in Access Gateway mode and E, EX_Port and F_Port on Fabric. Spaces are not allowed.

base64-string-type

Name	Type	Value	Description
base64-string-type	string	length: 0|4..max pattern: ([!-~]{4,})?	Base64 encoded string. Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string where each Base64 digit represents exactly 6 bits of data. So a plain string of 3 characters is represented by a Base64 encoded string of 4 characters. The generic string length relation between the plain and its encoded representation is: length of encoded string = 0 for null string or a multiple of 4 greater than or equal to (4 * n)/3, where 'n' is the number of characters in the plain string.

blade-state-type

Name	Type	Value	Description
blade-state-type	enumeration	enabled: The blade is enabled. disabled: The blade is disabled. inserted, not powered on(persistent): Blade is present but persistently powered off. inserted, not powered on: Blade is present but powered off. vacant: Blade is not present. powering up: Blade is powering up. inserted, not powered on(manual): Blade is present but manually poweredoff by the user. inserted, not powered on(not enough power): Blade is present but poweredoff due to not enough power. loading: Firmware is loading on the Blade. faulty: The blade is faulted. unknown: The blade state is unknown. diag running post1: Diag post1 is running on the blade. diag running post2: Diag post2 is running on the blade. initializing: Blade initialization is in progress.	The state of the blade.

blade-type-type

Name	Type	Value	Description
blade-type-type	enumeration	application blade: application blade type. core blade: core blade type. sw blade: switch blade type. cp blade: control processor blade type.	The type of the blade.

boot-up-self-test-status-type

Name	Type	Value	Description
boot-up-self-test-status-type	enumeration	pass: boot-up Self Tests passed fail: boot-up Self Tests failed not-run: boot-up Self Tests not yet run	Status of boot-up Self Tests

brocade-empty-string-type

Name	Type	Value	Description
brocade-empty-string-type	string	length: 0 pattern:	Definition of an empty string. This typedef is to be used within a union with other data types that cannot be modified to accommodate an empty string. The primary use case would be a node that utilizes a non-modifiable IETF typedef that does not account for an empty response. This type typedef should not be used on its own outside of a union.

brocade-hex-string-type

Name	Type	Value	Description
brocade-hex-string-type	string	length: 0..514 pattern: 0[xX][0-9a-fA-F]*	Brocade hexadecimal string format ---------- This type defines a generic case-insensitive hexadecimal string that requires a '0x' prefix (also case-insensitive). Modules that use this type may refine the length to a lower value, if needed.

brocade-telemetry-profile-operational-status-type

Name	Type	Value	Description
brocade-telemetry-profile-operational-status-type	enumeration	enabled: Profile is in enabled state. disabled: Profile is in disabled state. enabling: Profile is in enabling state. disabling: Profile is in disabling state. error: Profile is in error state.	Telemetry profile state.

brocade-telemetry-profile-state-type

Name	Type	Value	Description
brocade-telemetry-profile-state-type	enumeration	enable: Enable profile. disable: Disable profile.	Profile configuration state.

brocade-telemetry-profile-status-message-type

Name	Type	Value	Description
brocade-telemetry-profile-status-message-type	enumeration	success: Profile is enabled and no errors. schema-server-not-reachable: Profile is in error state due to schema server not reachable condition. stream-server-not-reachable: Profile is in error state due to stream server not reachable condition. kafka-internal-error: Profile is in error state due to kafka internal error. invalid-schema: Profile is in error state due to invalid schema installed on the switch. incompatible-schema: Profile is in error state due to incompatibile schema definition on the switch. server-connection-fatal-error: Profile is in error sttate due to abnormal connection termintation. ca-certificate-not-found: Profile is in error state due to no CA certificate found on the switch.	Telemetry profile status.

brocade-telemetry-protocol-type

Name	Type	Value	Description
brocade-telemetry-protocol-type	enumeration	plaintext: plaintext protocol. ssl: ssl protocol.	Telemetry protocol to use for stream server.

brocade-telemetry-server-status-type

Name	Type	Value	Description
brocade-telemetry-server-status-type	enumeration	init: Stream server is in init state. connecting: Stream server is in connecting state. connected: Stream server is in connected state. connection-error: Stream server is in connection error state. disconnected: Stream server is in disconnected state.	Telemetry stream server state.

brocade-telemetry-topic-type

Name	Type	Value	Description
brocade-telemetry-topic-type	enumeration	fiber-channel-port-metrics: Fiber Channel port metrics. flow-metrics: Flow Vision flow metrics. gige-port-metrics: Gigabit ethernet port metrics. eth-port-metrics: Ethernet port metrics. circuit-metrics: Fiber channel extension circuit metrics. tunnel-metrics: Fiber channel extension tunnel metrics. tunnel-qos-metrics: Fiber channel extension tunnel qos priority metrics. circuit-qos-metrics: Fiber channel extension circuit qos priority metrics. chassis-metrics: Chassis metrics like memory and cpu usage. fibrechannel-flow-violation-statistics: Flow Vision flow violation metrics.	Telemetry topics type.

cable-descriptor-type

Name	Type	Value	Description
cable-descriptor-type	enumeration	TW: A twin axial pair cable. TP: A twisted pair cable. MI: A miniature coaxial cable. TV: A video coaxial cable. M6: A multimode 62.5um cable. M5: A multimode 50um cable. OM3: A multimode 50m cable. SM: A singlemode fiber cable. MX: A multimode duplex cable.	The optical media cable type.

capable-speed-type

Name	Type	Value	Description
capable-speed-type	uint16	range: 1..1000	The supported transmission speeds of the media.

certificate-application-type

Name	Type	Value	Description
certificate-application-type	enumeration	commoncert: common certificate https: Certificate for HTTP secured radius: Certificate for Remote Authentication Dial-In User Service. ldap: Certificate for Lightweight Directory Access Protocol. syslog: Certificate for syslog fcap: Certificate for FCAP. RSA is the only supported key algorithm type. extension: Certificate for Extension.	Specifies the certificate type.

certificate-entity-type

Name	Type	Value	Description
certificate-entity-type	enumeration	cert: Specifies a switch certificate. ca: Specifies a CA certificate for extension only. ca-client: Specifies a CA client certificate. ca-server: Specifies a CA server certificate. csr: Specifies a CSR file.	Specifies the certificate for switch/ CA/CSR

certificate-hexdump-type

Name	Type	Value	Description
certificate-hexdump-type	string	length: 0|1..max pattern: [\r\n -~]*	The CSR, CA or switch certificate contents in raw hexadecimal format, including beginning and end markers, installed on the switch.

cfg-action-type

Name	Type	Value	Description
cfg-action-type	enumeration	save: Save the pending changes. disable: Disable the effective configuration. clear: Clear the defined configuration. Note that this stages the modifications in the defined-configuration. For the staged changes to become persistent, the user must save the changes using one of the following brocade-zone module leafs that reside under the effective-configuration container: 'cfg-action-v2': Specify 'save' to save the pending changes. 'cfg-name': Specify the name of the configuration to become enabled. It is recommended that the save be performed within the configured fabric-lock-timeout duration specified in the brocade-fibrechannel-configuration module under the zone-configuration container. transaction-abort: Abort the current transaction.	These action-type enums are to be used with the cfg-action-v2 leaf. The actions are performed against the zoning configuration.

chassis-distribute-domain-type

Name	Type	Value	Description
chassis-distribute-domain-type	union	fibrechannel:domain-id-type string	Input for distributing IP filter chassis wide in the specified switch in the fabric. Fabric ID or Domain ID. A wildcard (*) may be specified to include all switches in the fabric. Not Applicable for GET instance.

chassis-distribute-policy-type

Name	Type	Value	Description
chassis-distribute-policy-type	enumeration	ipfilter: IP filter policy database.	Policy names for chassis distribution.

chassis-user-friendly-name-type

Name	Type	Value	Description
chassis-user-friendly-name-type	string	length: 1..31 pattern: ([a-zA-Z][0-9a-zA-Z-._]{0,30})?	An ASCII name assigned to the switch chassis by the administrator.

circuit-admin-status-type

Name	Type	Value	Description
circuit-admin-status-type	enumeration	disabled: The admin status is disabled. enabled: The admin status is enabled. config: The admin status is config and in an internal state. ha-disabled: The admin-status is HA disabled. fenced: The admin-status is fenced. test: The admin-status is under test.	Extension circuit admin status type.

class-of-service-type

Name	Type	Value	Description
class-of-service-type	string	length: 0..256 pattern: (|bit-[4-9]|bit-[1-2][0-9]| bit-3[0-1]|class-f|class-1|class-2| class-3)*	A string representation of the Class of Service bit-field (see FC-SW). Bit 0 - Class F Represented as 'class-f' Bit 1 - Class 1 Represented as 'class-1' Bit 2 - Class 2 Represented as 'class-2' Bit 3 - Class 3 Represented as 'class-3' Bit 4 - obsolete Represented as 'bit-4' Bit 5: reserved Represented as 'bit-5' Bit 6 - obsolete Represented as 'bit-6' Bits 7 to 31: reserved Represented as 'bit-7' though 'bit-31', respectively

class-type

Name	Type	Value	Description
class-type	string	length: 2..13 pattern: zone|security|configuration| firmware|fabric|ls|cli|maps	Represents type of log need to be collected i.e. fabric related log etc.

clear-feature-type

Name	Type	Value	Description
clear-feature-type	enumeration	eula: The End User License Agreement for a switch.	Specifies the features to clear during the factory reset operation.

comments-type

Name	Type	Value	Description
comments-type	string	length: 0..256 pattern: ^$|Test failed|No SFP or chip support|See remote port results|Remote port is not ready to start the test|Too many offline events|Remote port test failed|Remote port WWN changed|Slot or port offline|Protocol error|Unable to start/restart the test|D-Port mode mismatch, Not D-Port|Module removed|No remote SFP or chip support|No remote port support or skipped|E_WRAP/O_WRAP set failed|User opted to skip|Test skipped due to SFP loopback error|Test timed out	Comments are additional textual information elaborating on the completion status when test is skipped or fails. It will return an empty string if no comments are available. The valid comments include: Test failed No SFP or chip support See remote port results Remote port is not ready to start the test Too many offline events Remote port test failed Remote port WWN changed Slot or port offline Protocol error Unable to start/restart the test D-Port mode mismatch, Not D-Port Module removed No remote SFP or chip support No remote port support or skipped E_WRAP/O_WRAP set failed User opted to skip Test skipped due to SFP loopback error Test timed out

compression-protocol-type

Name	Type	Value	Description
compression-protocol-type	enumeration	default: The compression is default. When this value is set for protocol level compression leaves i.e fc-compression-v2 and ip-compression-v2 leaf then the applicable compression level is determined by 'compression-tunnel' leaf value. disabled: The compression is disabled. deflate: The compression is set to deflate. aggressive-deflate: The compression is set to aggressive-deflate. fast-deflate: The compression is set to fast deflate. This value is only supported on SX6 and 7840.	Extension compression configurations supported at protocol level.

compression-tunnel-type

Name	Type	Value	Description
compression-tunnel-type	enumeration	disabled: The compression is disabled. deflate: The compression is set to deflate. aggressive-deflate: The compression is set to aggressive-deflate. fast-deflate: The compression is set to fast deflate. This value is only supported on SX6 and 7840.	Extension compression configurations supported at tunnel level.

congestion-detection-signal-cycles-type

Name	Type	Value	Description
congestion-detection-signal-cycles-type	uint16	range: 0|1..999	Signal threshold - congestion detection cycle count: 0 : Not Supported. 1 to 999 : Number of cycles at the indicated signaling scale used to detect Congestion Signals. see also 'congestion-detection-signal-scale-type'

congestion-detection-signal-scale-type

Name	Type	Value	Description
congestion-detection-signal-scale-type	enumeration	none: Congestion detection signals are not supported. seconds: Congestion Detection Signal Cycles in Seconds. milliseconds: Congestion Detection Signal Cycles in Milliseconds. microseconds: Congestion Detection Signal Cycles in Microseconds. nanoseconds: Congestion Detection Signal Cycles in Nanoseconds.	Signal threshold - congestion detection scale (unit of time). see also 'congestion-detection-signal-cycles-type'

congestion-detection-signal-type

Name	Type	Value	Description
congestion-detection-signal-type	enumeration	none: Congestion detection signals are not supported. warning: Warning congestion detection signal is supported. warning-and-alarm: Warning and Alarm congestion detection signals are supported.	Congestion detection signal capability.

congestion-state-type

Name	Type	Value	Description
congestion-state-type	enumeration	none: No congestion conditions reported credit-stalled: Device is exhibiting symptoms of BB credit stall lost-credit: Device is exhibiting symptoms of lost BB credit oversubscribed: Device is exhibiting symptoms of oversubscription	Congestion status.

connection-state-type

Name	Type	Value	Description
connection-state-type	enumeration	listen: Waiting for a connection request from remote TCP application. syn_sent: Waiting for an acknowledgment from the remote endpoint after having sent a connection request. syn_recv: This endpoint has received a connection request. established: Waiting for a connection request from a remote TCP application. fin_wait1: Waiting for a connection request from a remote TCP application. fin_wait2: Waiting for a connection request from a remote TCP application. time_wait: Waiting for a connection request from a remote TCP application. closed: Waiting for a connection request from a remote TCP application. close_wait: Waiting for a connection request from a remote TCP application. last_ack: The remote end has shut down, and the socket is closed. Waiting for an acknowledgement. closing: Waiting for a connection termination request acknowledgement from the remote TCP. unknown: The state of the connection is unknown	Specifies the state of the connection.

control-qos-type

Name	Type	Value	Description
control-qos-type	enumeration	priority-control: The 'control' qos represent the logical or virtual segregation of an extension tunnel to provide specific qos for control messaging and communication, including F-CLass traffic. This is also called as control tunnel as well and is common for both FC/IP protocols in case of hybrid tunnels. The control tunnel is a higher priority qos and has no guaranteed minimum bandwidth. As needed, it can use all available bandwidth up to the maximum configured bandwidth rate although this traffic utilizes very little bandwidth.	The control priority QoS support for extension-tunnel

counter32

Name	Type	Value	Description
counter32	uint32		The counter32 type represents a non-negative integer that monotonically increases until it reaches a maximum value of 2^32-1 (4294967295 decimal), when it wraps around and starts increasing again from zero. Counters have no defined 'initial' value, and thus, a single value of a counter has (in general) no information content. Discontinuities in the monotonically increasing value normally occur at re-initialization of the management system, and at other times as specified in the description of a schema node using this type. If such other times can occur, for example, the creation of a schema node of type counter32 at times other than re-initialization, then a corresponding schema node should be defined, with an appropriate type, to indicate the last discontinuity. The counter32 type should not be used for configuration schema nodes. A default statement SHOULD NOT be used in combination with the type counter32. In the value set and its semantics, this type is equivalent to the Counter32 type of the SMIv2.

counter64

Name	Type	Value	Description
counter64	uint64		The counter64 type represents a non-negative integer that monotonically increases until it reaches a maximum value of 2^64-1 (18446744073709551615 decimal), when it wraps around and starts increasing again from zero. Counters have no defined 'initial' value, and thus, a single value of a counter has (in general) no information content. Discontinuities in the monotonically increasing value normally occur at re-initialization of the management system, and at other times as specified in the description of a schema node using this type. If such other times can occur, for example, the creation of a schema node of type counter64 at times other than re-initialization, then a corresponding schema node should be defined, with an appropriate type, to indicate the last discontinuity. The counter64 type should not be used for configuration schema nodes. A default statement SHOULD NOT be used in combination with the type counter64. In the value set and its semantics, this type is equivalent to the Counter64 type of the SMIv2.

cp-name-type

Name	Type	Value	Description
cp-name-type	enumeration	cp0: CP in fixed-port switch and CP 0 in chassis. cp1: CP 1 in chassis.	Name of the host CP in fixed-port and chassis switch. Valid value for host CP in fixed-port switch is cp0. Valid values for host CP in chassis switch are cp0 and cp1.

credit-stall-state-type

Name	Type	Value	Description
credit-stall-state-type	enumeration	frame-loss: The frame loss state. perf-impact: The performance impact state. medium: The medium state. low: The low state. info: The informational state. no-congestion: The no congestion state. monitoring-paused: Monitoring is paused for the member.	The available states on the FPI credit-stall congestion conditions.

crypto-version-type

Name	Type	Value	Description
crypto-version-type	string	length: 1..16 pattern: [0-9a-zA-Z_.-]+	Specifies crypto version number.

current-type

Name	Type	Value	Description
current-type	decimal64	range: 0..999.99	The transceiver current of the media.

data-type-type

Name	Type	Value	Description
data-type-type	enumeration	unsigned-int32: The unsigned integer 32 bits. int32: The integer 32 bits data type. float: The float data type. unsigned-int64: The unsigned integer 64 bits type. enum: The enumeration type.	The data type support. Each monitoring system supports different data types for thresholds.

date-and-time

Name	Type	Value	Description
date-and-time	string	pattern: \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d+)?(Z|[\+\-]\d{2}:\d{2})	The date-and-time type is a profile of the ISO 8601 standard for representation of dates and times using the Gregorian calendar. The profile is defined by the date-time production in Section 5.6 of RFC 3339. The date-and-time type is compatible with the dateTime XML schema type with the following notable exceptions: (a) The date-and-time type does not allow negative years. (b) The date-and-time time-offset -00:00 indicates an unknown time zone (see RFC 3339) while -00:00 and +00:00 and Z all represent the same time zone in dateTime. (c) The canonical format (see below) of data-and-time values differs from the canonical format used by the dateTime XML schema type, which requires all times to be in UTC using the time-offset 'Z'. This type is not equivalent to the DateAndTime textual convention of the SMIv2 since RFC 3339 uses a different separator between full-date and full-time and provides higher resolution of time-secfrac. The canonical format for date-and-time values with a known time zone uses a numeric time zone offset that is calculated using the device's configured known offset to UTC time. A change of the device's offset to UTC time will cause date-and-time values to change accordingly. Such changes might happen periodically in case a server follows automatically daylight saving time (DST) time zone offset changes. The canonical format for date-and-time values with an unknown time zone (usually referring to the notion of local time) uses the time-offset -00:00.

date-code-type

Name	Type	Value	Description
date-code-type	string	length: 0..32 pattern: [-~]{0,32}	The vendor's manufacturing date code.

dcc-policy-member-type

Name	Type	Value	Description
dcc-policy-member-type	enumeration	switch: Switch member for DCC policy. device: Device member for DCC policy.	Identifies the WWN provided in operation is for switch or device.

default-control-type

Name	Type	Value	Description
default-control-type	enumeration	snmpv1: Defaulting the snmpv1 configurations. snmpv3: Defaulting the snmpv3 configurations. accesscontrol: Defaulting the access control configurations. deprecated: Please use ipfilter-policy list instead. system: Defaulting the system configurations. mibcap: Defaulting the snmp mib capability configurations. auditinterval: Defaulting the snmp audit interval. mibcapability: Defaulting the snmp mib capability configurations.	The SNMP default configuration options.

default-string-type

Name	Type	Value	Description
default-string-type	string	length: 1..max pattern: [-~]*	Default string

deskew-type

Name	Type	Value	Description
deskew-type	uint16	range: 0..255	Time in 10ns units. Used by trunking for fibrechannel ports.

dh-group-type

Name	Type	Value	Description
dh-group-type	enumeration	dh-group-null: Specifies NULL DH algorithm. dh-group-1024: Specifies DH group 1024. dh-group-1280: Specifies DH group 1280. dh-group-1536: Specifies DH group 1536. dh-group-2048: Specifies DH group 2048. dh-group-all: Specifies all the DH groups	Diffie-Hellman (DH) group.

diagnostics-operation-action-type

Name	Type	Value	Description
diagnostics-operation-action-type	enumeration	stop-test: This option will manually stop the test running on a diagnostic port (D_Port). This option is not supported for D_Port HBA connections. start-test: This option will manually start the test on a diagnostic port (D_Port). This option is not supported for D_Port HBA connections. restart-test: Re-starts the diagnostics test on a diagnostic port (D_Port) with configured parameters. This option is not supported for D_Port HBA connections.	Specifies the diagnostic operation to be performed on a D_Port.

discovered-mtu-type

Name	Type	Value	Description
discovered-mtu-type	uint32	range: 0|1280..9216	The discovered path maximum transmission unit for an interface. A value of '0' indicates MTU discovery is not enabled on the interface, or the circuit has not yet established a connection to the peer. The valid MTU values are from 1280-9216 when path MTU discovery is enabled and when circuit connection to peer is established.

distribute-policy-type

Name	Type	Value	Description
distribute-policy-type	enumeration	scc: Switch Connection Control policy. dcc: Device Connection Control policy. fcs: Fabric Configuration Server policy. password: Password policy. authentication: Fabric Element Authentication policy.	Policy names for distribution.

domain-id-type

Name	Type	Value	Description
domain-id-type	uint32	range: 0..239	Fibre Channel switch domain identifier. Value of zero indicates segmented link.

domain-index-type

Name	Type	Value	Description
domain-index-type	string	pattern: ([1-9]|[1-9][0-9]|1[0-9][0-9]|2[0123][0-9]),(([0-9])|([1-9][0-9])|([1-9][0-9][0-9])|([1-9][0-9][0-9][0-9])|([1-5][0-9][0-9][0-9][0-9])|(6[0-4][0-9][0-9][0-9])|(65[0-4][0-9][0-9])|(655[0-2][0-9])|(6553[0-5])|(-1))	Domain (D) and Index (I) zone members appear as 'D,I'. Domain is between 1 and 239; Index is between -1 and 65535.

domain-name

Name	Type	Value	Description
domain-name	string	length: 1..253 pattern: ((([a-zA-Z0-9_]([a-zA-Z0-9\-_]){0,61})?[a-zA-Z0-9]\.)*([a-zA-Z0-9_]([a-zA-Z0-9\-_]){0,61})?[a-zA-Z0-9]\.?)|\.	The domain-name type represents a DNS domain name. The name SHOULD be fully qualified whenever possible. Internet domain names are only loosely specified. Section 3.5 of RFC 1034 recommends a syntax (modified in Section 2.1 of RFC 1123). The pattern above is intended to allow for current practice in domain name use, and some possible future expansion. It is designed to hold various types of domain names, including names used for A or AAAA records (host names) and other records, such as SRV records. Note that Internet host names have a stricter syntax (described in RFC 952) than the DNS recommendations in RFCs 1034 and 1123, and that systems that want to store host names in schema nodes using the domain-name type are recommended to adhere to this stricter standard to ensure interoperability. The encoding of DNS names in the DNS protocol is limited to 255 characters. Since the encoding consists of labels prefixed by a length bytes and there is a trailing NULL byte, only 253 characters can appear in the textual dotted notation. The description clause of schema nodes using the domain-name type MUST describe when and how these names are resolved to IP addresses. Note that the resolution of a domain-name value may require to query multiple DNS records (e.g., A for IPv4 and AAAA for IPv6). The order of the resolution process and which DNS record takes precedence can either be defined explicitly or may depend on the configuration of the resolver. Domain-name values use the US-ASCII encoding. Their canonical format uses lowercase US-ASCII characters. Internationalized domain names MUST be A-labels as per RFC 5890.

dotted-quad

Name	Type	Value	Description
dotted-quad	string	pattern: ((\[0-9\]|\[1-9\]\[0-9\]|1\[0-9\]\[0-9\]|2\[0-4\]\[0-9\]|25\[0-5\])\\.){3}(\[0-9\]|\[1-9\]\[0-9\]|1\[0-9\]\[0-9\]|2\[0-4\]\[0-9\]|25\[0-5\])	An unsigned 32-bit number expressed in the dotted-quad notation, i.e., four octets written as decimal numbers and separated with the '.' (full stop) character.

dp-actstate-type

Name	Type	Value	Description
dp-actstate-type	enumeration	unavilable: The DP active state is unavailable. This is initial state and may be seen during bootup sequence of a DP. inactive: The DP state is inactive. This state indicates no active operations on the DP in progress. active: The DP state is active. all-complete: The DP state is all-complete. faulty: The DP state is faulty. invalid: The DP state is invalid.	The current active state of DP.

dp-dp-communication-type

Name	Type	Value	Description
dp-dp-communication-type	enumeration	unknown: The DP-DP communication status in not known at this moment. This is could be seen in initial bootup scenario when CP-DP connection itself is not established or up. not-configured: The DP-DP communication status in not programmed/configured. This could be seen in initial switch bootup scenario or slot poweroff/poweron sequence or HCL when CP-DP connection is established but programming of DP from CP is still pending or could be seen in some error scenarios. Also, in case of 7810 there is only one DP and hence not configured. up: The DP-DP communication status is established like applicable for SX6. down: The DP-DP communication status is not established, like applicable for SX6 blade in hybrid app-mode. not-applicable: The DP-DP communication status is not applicable. This can be seen in case of FCIP app-mode for example in SX6.	The DP to DP communication status.

dp-hcl-stage-type

Name	Type	Value	Description
dp-hcl-stage-type	enumeration	idle: The DP HA stage is idle. initializing: The DP HA stage is in initializing stage. preparing: The DP HCL stage is in preparing stage. FC-failover: The DP HCL stage is in FC failover stage. IP-failover: The DP HCL stage is in IP failover stage. reset: The DP HCL stage is in DP reset stage. wait: The DP HCL stage is in wait stage. FC-failback: The DP HCL stage is in FC failback stage. IP-failback: The DP HCL stage is in IP failback stage. DP-complete: The DP HCL stage is in DP complete stage. slot-complete: The DP HCL stage is in HCL complete stage. DP-done: The DP HCL stage is in DP done stage. VE-recovery: The DP HCL stage is in VE recovery stage. The VE recovery is done for VE which have encountered issues during HCL. simulator: The DP HCL stage is in simulator. This is an internal state for DP. invalid: The DP HCL stage is invalid stage and may be seen in error scenarios.	The different DP HCL related stage

dp-id-type

Name	Type	Value	Description
dp-id-type	uint16	range: 0..1	Extension Data Path Processor ID. Based on platform either it will have a single DP or dual DP. In case of single DP only DP0 is supported, and in case of dual DP both DP0 and DP1 are supported 0 : DP0 1 : DP1.

dp-state-type

Name	Type	Value	Description
dp-state-type	enumeration	uninitialized: The DP state is 'uninitialized'. This is initial state and may be seen during bootup sequence of a DP. initializing: The DP state is 'initializing'. This is initial state and may be seen during bootup sequence of a DP. pre-connecting: The DP state is in 'pre-connecting' stage of CP-DP communication. connecting: The DP state is in 'connecting' state and connection handshake of CP-DP communication in in progress. established: The DP state is in 'established' state i.e. connection handshake stage of CP-DP communication is completed. ready: The DP state is in 'ready' state. offline: The DP state is in 'offline' state i.e CP-DP connection is down. deferred-online: The DP state is in 'deferred-online' state. online: The DP state is in 'online' state. The config replay syncing of configuration from CP is complete. connection-close: The DP state is in 'connection-close' state. The CP-DP connection is going down. sync-failure: The DP state is in CP-DP config 'sync-failure' state. sync-in-progress: The DP state is in configuration 'sync-in-progress' state from CP. sync-complete: The DP state is in configuration 'sync-complete' state. reset: The DP is in 'reset' state. The DP is going for a reboot sequence. discovery: The DP is in 'discovery' state from CP. disconnecting: The DP is in 'disconnecting' state. faulty: The DP state is in faulty state. invalid: The DP state is in 'invalid' state.	The Different DP related state.

dscp

Name	Type	Value	Description
dscp	uint8	range: 0..63	The dscp type represents a Differentiated Services Code Point that may be used for marking packets in a traffic stream. In the value set and its semantics, this type is equivalent to the Dscp textual convention of the SMIv2.

dynamic-load-sharing-level-type

Name	Type	Value	Description
dynamic-load-sharing-level-type	enumeration	disabled: Disables the Dynamic Load Sharing (DLS) capability. dls: Enables only the base Dynamic Load Sharing (DLS) capability. lossless-dls: Enables the Dynamic Load Sharing (DLS) and LosslessDLS capability. two-hop-lossless-dls: Enables the Two-Hop LosslessDLS enhancement to LosslessDLS.	Supported combination for the Dynamic Load Sharing (DLS) capability

els-descriptor-type

Name	Type	Value	Description
els-descriptor-type	enumeration	none: No registered diagnostic ELS functions or descriptors. fpin-peer-congestion: FPIN ELS + Peer-Congestion descriptor registered. fpin-link-integrity: FPIN ELS + Link Integrity descriptor descriptor registered. fpin-congestion: FPIN ELS + Congestion descriptor descriptor registered. fpin-delivery: FPIN ELS + Delivery descriptor descriptor registered. pmn-decommission: PMN ELS + Inband port decommission descriptor registered. pmn-recommission: PMN ELS + Inband port recommission descriptor registered. pmn-fcsb-port-id: PMN ELS + FCSB port identification descriptor registered.	ELS + ELS-descriptor pairs that are exchanged via the Register Diagnostic Functions (RDF) ELS.

error-stats-type

Name	Type	Value	Description
error-stats-type	enumeration	none: Errors were not detected on the port. generic: Generic error encountered in test. enc-in: Encoding errors inside of frames. crc-err: Frames with CRC errors. trunc-frame: Frames shorter than minimum. frame-too-long: Frames longer than maximum. bad-eof: Frames with bad end-of-frame. enc-out: Encoding error outside of frames. bad-os: Invalid ordered set. c3-discard: Class 3 frames discarded. rx-c3-timeout: Class 3 receive frames discarded due to timeout. tx-c3-timeout: Class 3 transmit frames discarded due to timeout. unroutable: Frames that are unroutable. unreachable: Frames with unreachable destination. other-discard: Other C3 frame discards. link-failure: Link failures. sync-loss: Loss of synchronizations. signal-loss: Loss of signal. prim-seq-proto: Primitive sequence protocol error. inv-tx-word: Invalid transmit frames. invalid-crc: Crc error with good eof. delimiter-err: Frame delimiter errors. addr-id-err: Address ID errors. link-reset-in: Link resets received. link-reset-out: Link resets transmitted. port-ols-in: OLS primitives received. port-ols-out: OLS primitives transmitted. inv-arb: Invalid ARB. single-credit-loss: Single vcrdy/frame loss on link. multi-credit-loss: Multiple vcrdy/frame loss on link. fec-uncorrected: FEC uncorrected errors. pcs-block-error: PCS block errors. unknown-error: Maximum number of error stats.	Displays error stats on the port, if any.

eula-action-type

Name	Type	Value	Description
eula-action-type	enumeration	accept-eula: A confirmation of the user's acceptance of the EULA (Brocade End User License Agreement). Firmware download progresses only after you accept the EULA. decline-eula: A confirmation of the user's denial of the EULA. Firmware download aborts with an error when you decline the EULA. display-eula: Displays the latest EULA. The highest version of the EULA displays. Example: If switch is running FOS 9.0, and upgrading to FOS 9.0.1, then the 9.0.1 EULA displays. This principle also applies during downgrade.	The actions supported for EULA.

event-class-type

Name	Type	Value	Description
event-class-type	enumeration	zone: Log generated by zone module. security: Log generated by security module. configuration: Log generated by configuration event. firmware: Log generated by firmware event. fabric: Log generated by fabric event. ls: Log generated by logical switch module. cli: Log generated by CLI. maps: Log generated by MAPS module. ras: Log generated by RAS module. system: Log generated by system event.	Class type of event that generated log.

extension-certificate-entity-type

Name	Type	Value	Description
extension-certificate-entity-type	union	certificate-entity-type fibrechannel:brocade-empty-string-type	Specifies the extension certificate for switch/CA/CSR

extension-ipsec-profile-name

Name	Type	Value	Description
extension-ipsec-profile-name	enumeration	preshared: Specifies the 'preshared' extension IPsec policy name pki: Specifies the 'pki' extension IPsec policy name	Specifies the extension IPsec policy name.

extension-lan-interface-type

Name	Type	Value	Description
extension-lan-interface-type	union	extension-name-type brocade-interface-types:ge-interface-type	The Extension LAN interface type can be either a LAG name or else a GE port.

extension-name-type

Name	Type	Value	Description
extension-name-type	string	length: 1..31 pattern: ([a-zA-Z0-9~@%_+:\[\]])(([a-zA-Z0-9~@%_+:\[\]\-])|)[a-zA-Z0-9~@%_+:\[\]\-])*	The extension user defined name type definition.

extension-qos-type

Name	Type	Value	Description
extension-qos-type	union	control-qos-type fc-qos-type ip-qos-type	The control/FC/IP priority QoS support for extension-tunnel

fabric-id-type

Name	Type	Value	Description
fabric-id-type	uint32	range: 1..128	Fabric ID (FID)

fabric-incident-type

Name	Type	Value	Description
fabric-incident-type	enumeration	domain-id-forcibly-changed: The domain ID is forcibly changed. e-port-offline-transition: The E_Port transitioned to the offline state. reconfiguration: The fabric is reconfigured. segmentation: A fabric segmentation has occurred.	The events that occurred in a fabric.

fabric-wide-consistency-policy-type

Name	Type	Value	Description
fabric-wide-consistency-policy-type	enumeration	tolerant-scc: Switch Connection Control policy in tolerant mode. strict-scc: Switch Connection Control policy in strict mode. tolerant-dcc: Device Connection Control policy in tolerant mode. strict-dcc: Device Connection Control policy in strict mode. tolerant-fcs: Fabric Configuration Server policy in tolerant mode. strict-fcs: Fabric Configuration Server policy in strict mode. clear: Empty policy. Clears out all existing fabric wide set policy This should not be specified with other policies.	Policy types for fabric wide consistency policy.

fc-lag-link-status-type

Name	Type	Value	Description
fc-lag-link-status-type	enumeration	up: FC-LAG member port is operationally up. down: FC-LAG member port is operationally down.	Supported FC-LAG and member ports link status types:

fc-qos-type

Name	Type	Value	Description
fc-qos-type	enumeration	FC-priority-high: The 'FC-high' qos represent the logical or virtual segregation of an extension tunnel to provide highest priority QoS for FC data traffic. This WAN optimized TCP traffic class is via per priority TCP QoS. FC-priority-medium: The 'FC-medium' qos represent the logical or virtual segregation of an extension tunnel to provide medium priority QoS for FC data traffic. This WAN optimized TCP traffic class is via per priority TCP QoS. FC-priority-low: The 'FC-low' qos represent the logical or virtual segregation of an extension tunnel to provide low priority QoS for FC data traffic. This WAN optimized TCP traffic class is via per priority TCP QoS.	The FC priority QoS support for extension-tunnel

fc-speed-type

Name	Type	Value	Description
fc-speed-type	uint64	range: 0|1000000000|2000000000| 4000000000|8000000000| 10000000000|16000000000| 32000000000|53000000000| 64000000000	The current bandwidth of the port. 0 : Auto negotiated. 1000000000 : 1 GFC 2000000000 : 2 GFC 4000000000 : 4 GFC 8000000000 : 8 GFC 10000000000 : 10 GFC 16000000000 : 16 GFC 32000000000 : 32 GFC 53000000000 : 53 GFC 64000000000 : 64 GFC

fc4-features-type

Name	Type	Value	Description
fc4-features-type	string	length: 0..512 pattern: (|FCP-Initiator|FCP-Target| FC-NVMe-Discovery-Service|FC- NVMe-Initiator|FC-NVMe-Target)*	The registered FC-4 Features in relation to a given FC-4 Type.

fc4-type-type

Name	Type	Value	Description
fc4-type-type	string	length: 0..512 pattern: (0[xX][0-9a-fA-F]{2}| | IPFC|FCP|FCP-Features|SATA- Tunnel|SBCCS|SBCCS-Channel| SBCCS-Control-Unit|FC-CT|FC-SW| FC-IFR|FC-NVMe|HIPPI-FP|MIL- STD-1553|ASM|FC-VI|Application- Services|Generic-FC-Features|RNID- Topology-Discovery)*	A space-delimited string of FC-4 types: IPFC (0x05) FCP (0x08) FCP-Features (0x0A) SATA-Tunnel (0x14) SBCCS (0x18) SBCCS-Channel (0x1B) SBCCS-Control-Unit (0x1C) FC-CT (0x20) FC-SW (0x22) FC-IFR (0x25) FC-NVMe (0x28) HIPPI-FP (0x40) MIL-STD-1553 (0x48) ASM (0x49) FC-VI (0x58) Application-Services (0x60) Generic-FC-Features (0xDE) RNID-Topology-Discovery (0xDF) 0xnn (Unknown/Reserved <hexadecimal value>)

fcid-hex-string-type

Name	Type	Value	Description
fcid-hex-string-type	string	length: 8 pattern: 0[xX][0-9a-fA-F]*	Definition of the Fibre Channel address for an Nx_Port (FCID), hexadecimal format. The format allows for six hexadecimal digits prefixed by '0x' or '0X'. For example: 0x010200

fcid-type

Name	Type	Value	Description
fcid-type	uint32	range: 0..16777215	Fibre channel ID (FCID), decimal format. Deprecated: Please use 'fcid-hex-string-type' instead.

fcr-device-state

Name	Type	Value	Description
fcr-device-state	enumeration	configured: Device is configured to be in an LSAN, but the device is neither imported nor exists in this fabric. initializing: Device is in an intermediate state. It is not yet imported into the fabric. exist: Device exists in this fabric (the fabric of the zone entry). imported: Device has been imported (proxy created) into this fabric.	State information of the device w.r.t FCR, valid states include are configured, initializing, exist, imported.

fec-type

Name	Type	Value	Description
fec-type	enumeration	off: No Forward Error Correction clause configured on gigabitethernet interface. cl108: Forward Error Correction clause is CL108. Valid for GE port at 25-ge protocol-speed. cl91: Forward Error Correction clause is CL91. Valid for GE port at 100-ge protocol-speed.	Specifies the configured Forward Error Correction clause of gigabitethernet interfaces.

fibre-channel-router-owner

Name	Type	Value	Description
fibre-channel-router-owner	enumeration	local: The owner front domain belongs to the current fibrechannel router. remote: The owner front domain belongs to the other fibrechannel router. remote-fabric: The owner front domain belongs to the fibrechannel router in another Backbone fabric. not-applicable: The owner fibrechannel router status is not applicable.	Represents the owner fibrechannel router of the front domain.

fibrechannel-trunk-type

Name	Type	Value	Description
fibrechannel-trunk-type	enumeration	inter-switch-link: trunk is a logical bundle of e-ports on a port blade or pizza box i.e. switch to switch links. inter-chassis-link: trunk is a logical bundle of e-ports on a core blade in a chassis i.e. chassis to chassis links. inter-fabric-link: trunk is a logical bundle of ex-ports between two fabrics using port blades or pizza box. f-port: trunk is a logical bundle of f-ports i.e. switch to device links on port blade or pizza box. f-port-access-gateway: trunk is a logical bundle of switch to switch links in a access gateway mode on port blades or pizza box.	Types of links which have been bundled together in a trunk.

firmware-version-type

Name	Type	Value	Description
firmware-version-type	string	length: 1..255 pattern: [-~]{1,255}	A human readable string identifying the firmware version running on the switch.

fru-date-type

Name	Type	Value	Description
fru-date-type	string	length: 0|27|28|29|30 pattern: (Day: (([ ][1-9])|([1-2][0-9])|([3][0-1])|([ ][0])) Month: (([ ][1-9])|([1][0-2])|([ ][0])) Year: (([2][0-9]{3})|([1-9][0-9]*)|([0])))?	Indicates the FRU date type.

fru-state-type

Name	Type	Value	Description
fru-state-type	enumeration	faulty: The faulty state. in: The plug-in state. out: The plug-out state. off: The power off state. on: The power on state.	The FRU states.

fru-type

Name	Type	Value	Description
fru-type	enumeration	sw blade: Indicates fru is switch blade. cp blade: Indicates fru is cp blade. core blade: Indicates fru is core blade. application blade: Indicates fru is application blade. fan: Indicates fru is fan. wwn: Indicates fru is wwn card. power supply: Indicates fru is power supply. unknown: Indicates unknown fru.	It indicates the fru type.

gauge32

Name	Type	Value	Description
gauge32	uint32		The gauge32 type represents a non-negative integer, which may increase or decrease, but shall never exceed a maximum value, nor fall below a minimum value. The maximum value cannot be greater than 2^32-1 (4294967295 decimal), and the minimum value cannot be smaller than 0. The value of a gauge32 has its maximum value whenever the information being modeled is greater than or equal to its maximum value, and has its minimum value whenever the information being modeled is smaller than or equal to its minimum value. If the information being modeled subsequently decreases below (increases above) the maximum (minimum) value, the gauge32 also decreases (increases). In the value set and its semantics, this type is equivalent to the Gauge32 type of the SMIv2.

gauge64

Name	Type	Value	Description
gauge64	uint64		The gauge64 type represents a non-negative integer, which may increase or decrease, but shall never exceed a maximum value, nor fall below a minimum value. The maximum value cannot be greater than 2^64-1 (18446744073709551615), and the minimum value cannot be smaller than 0. The value of a gauge64 has its maximum value whenever the information being modeled is greater than or equal to its maximum value, and has its minimum value whenever the information being modeled is smaller than or equal to its minimum value. If the information being modeled subsequently decreases below (increases above) the maximum (minimum) value, the gauge64 also decreases (increases). In the value set and its semantics, this type is equivalent to the CounterBasedGauge64 SMIv2 textual convention defined in RFC 2856

ge-interface-type

Name	Type	Value	Description
ge-interface-type	string	pattern: (([1][0-2]|[0-9])/(1[0-7]|[0-9]))	GE interface of an Extension blade or system. The slot and port number of the port (slot/port).

ge-protocol-type

Name	Type	Value	Description
ge-protocol-type	enumeration	LAN: The gigabitethernet is used for LAN side connections. WAN: The gigabitethernet is used for WAN side connections.	The different GE protocol settings.

gen-certificate-entity-type

Name	Type	Value	Description
gen-certificate-entity-type	enumeration	cert: To Generate a switch certificate. csr: To generate a CSR.	Specifies the type for certificate generation.

group-name

Name	Type	Value	Description
group-name	enumeration	ro: Indicates the SNMP read only group name. deprecated: Please use 'read-only' instead. rw: Indicates the SNMP read write group name. deprecated: Please use 'read-write' instead. read-only: Indicates the SNMP read only group name. read-write: Indicates the SNMP read write group name.	It indicates the SNMP group name

hash-algorithm-type

Name	Type	Value	Description
hash-algorithm-type	enumeration	md5: md5 hash type. sha256: sha256 hash type. sha512: sha512 hash type.	Sets the hash type.

hcl-status-type

Name	Type	Value	Description
hcl-status-type	enumeration	unavailable: The DP HCL status is 'unavailable'. ready: The DP HCL status is 'ready'. HCL-notification-received: The DP HCL status is in HCL notification received. HCL-pending: The DP HCL status is in 'HCL-pending' state. The CP-DP version is not same and DP HCL is not started. HCL-in-progress: The DP HCL status is in 'HCL-in-progress' state. firmwaredownload-in-progress: The DP HCL status is in 'firmwaredownload-in-progress' state. booting: The DP HCL status is in 'booting' state. faulty: The DP HCL status is in 'faulty' state. dp-communication-down: The DP HCL status is 'dp-communication-down' state. HCL-unsupported: The DP HCL status is 'HCL-unsupported' state. invalid: The DP HCL status is in 'invalid' state.	The different hot code Load status

hex-string

Name	Type	Value	Description
hex-string	string	pattern: ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2})*)?	A hexadecimal string with octets represented as hex digits separated by colons. The canonical representation uses lowercase characters.

home-virtual-fabric-type

Name	Type	Value	Description
home-virtual-fabric-type	union	fibrechannel:fabric-id-type uint32	Specifies the account's home Virtual Fabric. The valid values for home-virtual-fabric-id is in the range '1-128'. A value of '0' may be used in case the home-virtual-fabric-id is not set or not valid.

host

Name	Type	Value	Description
host	union	inet:ip-address inet:domain-name	The host type represents either an IP address or a DNS domain name.

host-type

Name	Type	Value	Description
host-type	union	inet:ip-address inet:ip-prefix string	Specifies the IP address.

interface-type

Name	Type	Value	Description
interface-type	string	length: 6..16 pattern: ((fc|ge|te)/([0-9]|1[0-9])/([1][0-9][0-9][0-9]|[0-9][0-9][0-9][0-9]|[0-9][0-9]|[0-9]))	The network interface name in 3-tuple canonical format: interface-id/slot/port. The interface IDs include fc, ge, and te.

ip-address

Name	Type	Value	Description
ip-address	union	inet:ipv4-address inet:ipv6-address	The ip-address type represents an IP address and is IP version neutral. The format of the textual representation implies the IP version. This type supports scoped addresses by allowing zone identifiers in the address format.

ip-address-no-zone

Name	Type	Value	Description
ip-address-no-zone	union	inet:ipv4-address-no-zone inet:ipv6-address-no-zone	The ip-address-no-zone type represents an IP address and is IP version neutral. The format of the textual representation implies the IP version. This type does not support scoped addresses since it does not allow zone identifiers in the address format.

ip-extension-interface-type

Name	Type	Value	Description
ip-extension-interface-type	union	lan-interface-type ge-interface-type	Different extension interface type supported for IP Extension.

ip-gateway-type

Name	Type	Value	Description
ip-gateway-type	union	inet:ip-address string	Specifies the IP address of an IP router that can route packets to the destination IP address. This attribute can contain '*' while retrieving route entries which are created internally by the system.

ip-prefix

Name	Type	Value	Description
ip-prefix	union	inet:ipv4-prefix inet:ipv6-prefix	The ip-prefix type represents an IP prefix and is IP version neutral. The format of the textual representations implies the IP version.

ip-prefix-length

Name	Type	Value	Description
ip-prefix-length	union	ipv4-prefix-length ipv6-prefix-length	The prefix length operator for the IPv4/IPv6 address.

ip-qos-type

Name	Type	Value	Description
ip-qos-type	enumeration	IP-priority-high: The 'IP-high' qos represent the logical or virtual segregation of an extension tunnel to provide highest priority QoS for IP LAN data traffic. This WAN optimized TCP traffic class is via per priority TCP QoS. This is present only when the tunnel is a hybrid tunnel. IP-priority-medium: The 'IP-medium' qos represent the logical and or physical segregation of an extension tunnel to provide medium priority QoS for IP LAN data traffic. This WAN optimized TCP traffic class is via per priority TCP QoS. This is present only when the tunnel is a hybrid tunnel. IP-priority-low: The 'IP-low' qos represent the logical or virtual segregation of an extension tunnel to provide low priority QoS for IP LAN data traffic. This WAN optimized TCP traffic class is via per priority TCP QoS. This is present only when the tunnel is a hybrid tunnel.	The IP priority QoS support for extension-tunnel

ip-version

Name	Type	Value	Description
ip-version	enumeration	unknown: An unknown or unspecified version of the Internet protocol. ipv4: The IPv4 protocol as defined in RFC 791. ipv6: The IPv6 protocol as defined in RFC 2460.	This value represents the version of the IP protocol. In the value set and its semantics, this type is equivalent to the InetVersion textual convention of the SMIv2.

ipfilter-action-type

Name	Type	Value	Description
ipfilter-action-type	enumeration	activate: Activates the specified IP filter policy. clone: Creates a replica of an existing IP filter policy.	Indicates the action that must be taken on IPFilter policies.

ipfilter-ip-version-type

Name	Type	Value	Description
ipfilter-ip-version-type	enumeration	IPv4: Creates an IP filter policy with the specified type IPv4. IPv6: Creates an IP filter policy with the specified type IPv6.	Indicates the type of policy IPv4 or IPv6.

ipfilter-name-type

Name	Type	Value	Description
ipfilter-name-type	string	length: 1..20 pattern: ([a-zA-Z0-9_]{1,20})	The name of the ipfilter policy.

ipfilter-operation-action-type

Name	Type	Value	Description
ipfilter-operation-action-type	enumeration	default: Resets the ipfilter policies. Activates default policies, deletes all the custom policies.	Indicates the operation action that must be taken on IPFilter policies.

ipfilter-permission-type

Name	Type	Value	Description
ipfilter-permission-type	enumeration	permit: Specifies the permit action associated with this rule. deny: Specifies the deny action associated with this rule.	Blocking or permitting the ports.

ipfilter-protocol-type

Name	Type	Value	Description
ipfilter-protocol-type	enumeration	tcp: Specifies the protocol type tcp udp: Specifies the protocol type udp	Types of protocol TCP or UDP.

ipfilter-traffic-type

Name	Type	Value	Description
ipfilter-traffic-type	enumeration	INPUT: Specifies the type of traffic INPUT. FORWARD: Specifies the type of traffic FORWARD.	Specifying the type of traffic that are allowed for the specified IP address.

ipv4-address

Name	Type	Value	Description
ipv4-address	string	pattern: (([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(%[\p{N}\p{L}]+)?	The ipv4-address type represents an IPv4 address in dotted-quad notation. The IPv4 address may include a zone index, separated by a % sign. The zone index is used to disambiguate identical address values. For link-local addresses, the zone index will typically be the interface index number or the name of an interface. If the zone index is not present, the default zone of the device will be used. The canonical format for the zone index is the numerical format

ipv4-address-no-zone

Name	Type	Value	Description
ipv4-address-no-zone	inet:ipv4-address	pattern: [0-9\.]*	An IPv4 address without a zone index. This type, derived from ipv4-address, may be used in situations where the zone is known from the context and hence no zone index is needed.

ipv4-prefix

Name	Type	Value	Description
ipv4-prefix	string	pattern: (([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])/(([0-9])|([1-2][0-9])|(3[0-2]))	The ipv4-prefix type represents an IPv4 address prefix. The prefix length is given by the number following the slash character and must be less than or equal to 32. A prefix length value of n corresponds to an IP address mask that has n contiguous 1-bits from the most significant bit (MSB) and all other bits set to 0. The canonical format of an IPv4 prefix has all bits of the IPv4 address set to zero that are not part of the IPv4 prefix.

ipv4-prefix-length

Name	Type	Value	Description
ipv4-prefix-length	uint16	range: 1..32	The prefix length operator for the IPv4 address.

ipv6-address

Name	Type	Value	Description
ipv6-address	string	pattern: ((:|[0-9a-fA-F]{0,4}):)([0-9a-fA-F]{0,4}:){0,5}((([0-9a-fA-F]{0,4}:)?(:|[0-9a-fA-F]{0,4}))|(((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])))(%[\p{N}\p{L}]+)? pattern: (([^:]+:){6}(([^:]+:[^:]+)|(.*\..*))|((([^:]+:)*[^:]+)?::(([^:]+:)*[^:]+)?)(%.+)?	The ipv6-address type represents an IPv6 address in full, mixed, shortened, and shortened-mixed notation. The IPv6 address may include a zone index, separated by a % sign. The zone index is used to disambiguate identical address values. For link-local addresses, the zone index will typically be the interface index number or the name of an interface. If the zone index is not present, the default zone of the device will be used. The canonical format of IPv6 addresses uses the textual representation defined in Section 4 of RFC 5952. The canonical format for the zone index is the numerical format as described in Section 11.2 of RFC 4007.

ipv6-address-no-zone

Name	Type	Value	Description
ipv6-address-no-zone	inet:ipv6-address	pattern: [0-9a-fA-F:\.]*	An IPv6 address without a zone index. This type, derived from ipv6-address, may be used in situations where the zone is known from the context and hence no zone index is needed.

ipv6-flow-label

Name	Type	Value	Description
ipv6-flow-label	uint32	range: 0..1048575	The ipv6-flow-label type represents the flow identifier or Flow Label in an IPv6 packet header that may be used to discriminate traffic flows. In the value set and its semantics, this type is equivalent to the IPv6FlowLabel textual convention of the SMIv2.

ipv6-prefix

Name	Type	Value	Description
ipv6-prefix	string	pattern: ((:|[0-9a-fA-F]{0,4}):)([0-9a-fA-F]{0,4}:){0,5}((([0-9a-fA-F]{0,4}:)?(:|[0-9a-fA-F]{0,4}))|(((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])))(/(([0-9])|([0-9]{2})|(1[0-1][0-9])|(12[0-8]))) pattern: (([^:]+:){6}(([^:]+:[^:]+)|(.*\..*)))|((([^:]+:)*[^:]+)?::(([^:]+:)*[^:]+)?)(/.+)	The ipv6-prefix type represents an IPv6 address prefix. The prefix length is given by the number following the slash character and must be less than or equal to 128. A prefix length value of n corresponds to an IP address mask that has n contiguous 1-bits from the most significant bit (MSB) and all other bits set to 0. The IPv6 address should have all bits that do not belong to the prefix set to zero. The canonical format of an IPv6 prefix has all bits of the IPv6 address set to zero that are not part of the IPv6 prefix. Furthermore, the IPv6 address is represented as defined in Section 4 of RFC 5952.

ipv6-prefix-length

Name	Type	Value	Description
ipv6-prefix-length	uint16	range: 1..128	The prefix length operator for the IPv6 address.

keep-feature-type

Name	Type	Value	Description
keep-feature-type	enumeration	ip: The IP address of the switch.	The specified feature is not cleared during the factory reset operation.

keysize-type

Name	Type	Value	Description
keysize-type	enumeration	1024: key size of 1024 bits 2048: key size of 2048 bits 4096: key size of 4096 bits. Not supported by FCAP. Generation of a 4096-bit key is not supported via the REST API. 8192: key size of 8192 bits. Not supported by FCAP. Key-pair generation with 8192 bits is highly CPU intensive. Can cause high CPU usage and session time out. On such instance, user is advised to verify the generated status by another GET instance. Generation of a 8192-bit key is not supported via the REST API. P384: elliptic 384 bit curve P256: elliptic 256 bit curve P521: elliptic 521 bit curve 0: key size of 0 bits. No key available.	Specifies the size of the key. The greater the value, the more secure is the connection; however, performance degrades with size.

known-host-type

Name	Type	Value	Description
known-host-type	string	length: 1..299 pattern: [-~]*	The typedef for sshutil known-host-type which is of the format [host]:port-number, the host type includes both IPv4/IPv6 address formats and also fully qualified domain name. The port number includes only port number till 65535.

l4-common-protocol

Name	Type	Value	Description
l4-common-protocol	enumeration	any: The L4 Protocol is any, in case of filtering any L4 protocol value would be matched. TCP: The L4 Protocol is TCP. UDP: The L4 Protocol is UDP. ICMP: The L4 Protocol is ICMP. ICMP6: The L4 Protocol is ICMP6. VRRP: The L4 Protocol is VRRP.	The common L4 protocol strings

l4-protocol-number

Name	Type	Value	Description
l4-protocol-number	uint32	range: 0..255	The L4 protocol number which uniquely identifies the L4 protocol. The valid value for L4 protocol number is from 0-255. example

l4-protocol-type

Name	Type	Value	Description
l4-protocol-type	union	l4-protocol-number l4-common-protocol	The different L4 protocol values supported either its a protocol number from 0-255 or a known protocol string.

lag-port-timeout

Name	Type	Value	Description
lag-port-timeout	enumeration	short: The short member port timeout timeout value is 3 sec long: The long member port timeout timeout value is 90 sec	The portchannel member can have short or long timeout configured.

lan-interface-type

Name	Type	Value	Description
lan-interface-type	string	pattern: ([1][0-2][0-9])/LAN	Virtual LAN SVI port interface corresponding to a blade or switch.

ldap-tls-mode-type

Name	Type	Value	Description
ldap-tls-mode-type	enumeration	starttls: Selects StartTLS mode ldaps: Selects LDAPS mode	LDAP TLS mode of StartTLS or LDAPS.

link-fault-descriptor-group

Description

Link Fault Capabilities descriptor group:

The Link Fault Capabilities descriptor represents the Link Degrade Signaling parameters for a given N_Port or F_Port. See FC-FS-5 (Reed-Solomon decoder) for more information.

Parameters

Table 232: link-fault-descriptor-group

Name	Type	Description
degrade-activate-threshold	uint32	The value of a 32-bit register that specifies a symbol error count. The value here controls the threshold used to activate the Remote Degrade Bit being sent in the Alignment Marker field.
degrade-deactivate-threshold	uint32	The value of a 32-bit register that specifies a symbol error count. The value here controls the threshold used to deactivate the Remote Degrade Bit being sent in the Alignment Marker field.
fec-degrade-interval	uint32	The value of a 32-bit register that specifies the number of RS-FEC code words that make up a Degrade Interval.

lldp-mandatory-tlv-name-type

Name	Type	Value	Description
lldp-mandatory-tlv-name-type	enumeration	chassis-id: Chassis ID TLV port-id: Port ID TLV time-to-live: Time To Live TLV	LLDP Mandatory TLV names

lldp-neighbor-parameters-group

Description

This group contains the parameters for the LLDP neighbor device on the given interface

Parameters

Table 233: lldp-neighbor-parameters-group

Name	Type	Description
dead-interval	uint32	The dead interval of the LLDP neighbor.
remaining-life	uint32	The remaining life of the LLDP neighbor.
remote-interface-name	string	The remote interface name of the LLDP neighbor device connected to local switch ethernet interface.
chassis-id	string	The chassis ID of the LLDP neighbor.
system-name	string	The system name of the LLDP neighbor.

lldp-optional-tlv-name-type

Name	Type	Value	Description
lldp-optional-tlv-name-type	enumeration	dcbx: IEEE Data Center Bridging eXchange TLV fcoe-app: IEEE DCBx FCoE Application TLV fcoe-lls: IEEE DCBx FCoE Logical Link TLV dot1: IEEE 802.1 Organizationally Specific TLV dot3: IEEE 802.1 Organizationally Specific TLV mgmt-addr: Management Address TLV port-desc: Port Description TLV sys-cap: System Capabilities TLV sys-desc: System Description TLV sys-name: System Name TLV	LLDP Optional TLV names

lldp-statistics-parameters-group

Description

This group contains the LLDP statistics on the given interface

Parameters

Table 234: lldp-statistics-parameters-group

Name	Type	Description
out-frames	ietf:zero-based-counter64	The number of LLDP out frames.
in-frames	ietf:zero-based-counter64	The number of LLDP In frames.
frames-aged-out	ietf:zero-based-counter64	The number of LLDP aged out frames.
frames-discarded	ietf:zero-based-counter64	The number of LLDP discarded frames.

Name	Type	Description
error-frames	ietf:zero-based-counter64	The number of LLDP error frames
tlv-discarded	ietf:zero-based-counter64	The number of discarded LLDP TLVs.
tlv-unrecognized	ietf:zero-based-counter64	The number of unrecognized LLDP TLVs.

logical-path-operational-states

Name	Type	Value	Description
logical-path-operational-states	enumeration	Oper: A logical path is in operational state. Reset: A logical path is in reset state.	Describes the set of operational states for a logical path

long-distance-type

Name	Type	Value	Description
long-distance-type	enumeration	disabled: Long distance is disabled for this port. l0: l0 configures the port as a regular port. l1: l1 configures the port as long (<= 50 km). l2: l2 configures the port as super long (<= 100km). le: le configures an E_Port distance as greater than 5 km and up to 10 km. lm: lm configures the port as medium long (<= 25 km). ld: ld configures the value as automatic long distance. ls: ls mode configures the value as a static long distance link with a fixed buffer allocation greater than 10 km.	Supported modes for long distance configuration of a port.

los-tov-mode-type

Name	Type	Value	Description
los-tov-mode-type	enumeration	disabled: LOS_TOV mode is disabled on the port. enabled-fixed: LOS_TOV mode is enabled on fixed-speed ports. enabled-all: LOS_TOV mode is enabled on both fixed-speed and auto-negotiated ports. invalid: LOS_TOV mode is not valid for this port.	Supported modes of los-tov for a port.

mac-address

Name	Type	Value	Description
mac-address	string	pattern: [0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5}	The mac-address type represents an IEEE 802 MAC address. The canonical representation uses lowercase characters. In the value set and its semantics, this type is equivalent to the MacAddress textual convention of the SMIv2.

management-port-connection-group

Description

Management ethernet interface connection details.

Parameters

Table 235: management-port-connection-group

Name	Type	Description
cp-name	fibrechannel:cp-name-type	Specifies the name of the CP to which the connection belongs to.
ip-family	inet:ip-version	Specifies the internet protocol version that connection belongs to.
operational-status	enumeration	Specifies if the connection is currently active or not.
local-ip-address	inet:ip-address	Specifies the local/source IPv4/IPv6 address of the network connection.
local-port	inet:port-number	Specifies the local/source port number of the network connection.
remote-ip-address	inet:ip-address	Specifies the remote/destination IPv4/IPv6 address of the network connection.
remote-port	inet:port-number	Specifies the remote/destination port number of the network connection.
start-time	yang:date-and-time	Timestamp when the socket connection was opened at the switch.
state-update-time	union {yang:date-and-time;fibrechannel:brocade-empty-string-type;}	Timestamp when the socket connection state changed.
protocol	string	Represents the transport connection protocol used.
connection-state	brocade-interface-types:connection-state-type	State of the connection.
process-name	string	Name of the process that opened the socket.
process-id	uint16	Process id of the program that is responsible for the connection.

manufacturer-type

Name	Type	Value	Description
manufacturer-type	string	length: 1..63 pattern: [-~]*	The name of the organization responsible for producing the blade.

maps-dashboard-category-type

Name	Type	Value	Description
maps-dashboard-category-type	enumeration	port-health: The Port Health category monitors port statistics and takes action based on the configuration thresholds and actions. You can configure thresholds per port type and apply the configuration to all ports of the specified type. Ports whose thresholds can be monitored include physical ports, D_Ports, E_Ports, and F_Ports. The Port Health category also monitors the physical aspects of a small form-factor pluggable (SFP) transceiver, such as voltage, current, receive power (RXP), and transmit power (TXP) in physical ports, D_Ports, E_Ports, and F_Ports. backend-port-health: The Back-End port health category monitors the health of Back-End switch ports such as CRC, link reset errors, frame length (either too long or truncated), ITW, and so on. extension-ge-port-health: The Gigabit Ethernet (GE) ports category monitors statistics for GE ports and takes action based on the configuration thresholds and actions. You can configure thresholds and apply the configure to all ports. security-violations: The Security Health category monitors security violations on the switch and takes action based on the configure thresholds and their actions. fabric-state-changes: The Fabric State Changes category contains areas of potential fabric- related or switch-related problems, such as zone changes, fabric segmentation, E_Port down, fabric reconfiguration, domain ID changes, and fabric logins. fru-health: The FRU Health category enables you to define rules for field replaceable units (FRUs). extension-health: The Extension Health category enables you to define rules for Extension health, including circuit state changes, circuit state utilization, and packet loss. switch-resources: The Switch Resource category enables you to monitor your system's temperature, flash usage, memory usage, and CPU. fabric-performance-impact: The FPI category monitors parameters such as congestion due to credit-stall, oversubscription, and so on. io-latency: The IO Latency category	The dashboard category. Typically, each rule belongs to a dashboard category.

maps-data-unit

Name	Type	Value	Description
maps-data-unit	enumeration	CRCs: The CRC errors. ITWs: The ITWs errors. timeouts: The Class 3 frames discard due to timeout. milli-ampere: The milli ampere current. days: The days. errors: The errors. loss-of-signals: The loss of signals. loss-of-synchronizations: The loss of synchronization. violations: The violations. ports: The ports. extension-flows: The IP extension flows. it-flows: The target-to-host or vice versa ratio. certificates: The certificates. segmentations: The total times the port or fabric segmented from the fabric. changes: The changes. logins: The number of logins. IOs: The IOs. IOPS: The IOs per second. MBps: The bandwidth in mega bytes per second. %: The percentage value. hours: The hours. centigrade: The centigrade. bytes: The bytes. micro-seconds: The micro seconds. milli-seconds: The milli seconds. milli-volts: The milli volts. milli-amps: The milli ampere. micro-watts: The micro watts. aborts: The SCSI aborts count. reserves: The SCSI reserves. not-applicable: Not applicable. LFs: The link failure errors. loss-of-signal: The loss of signal errors. LRs: The LRs errors. routing-errors: The routing errors. events: The number of events. nvme-sler: The NVME sequence level error recovery. io-errors: The number of SCSI or NVMe IO errors.	The unit of the monitoring system.

maps-db-date-type

Name	Type	Value	Description
maps-db-date-type	string	length: 0..100 pattern: current|[0-9\-]{2}/[0-9\-]{2}/[0-9\-]{2}	The date in MM/DD/YY format and the string 'current'. Where 'current' indicates the current day.

maps-event-severity-type

Name	Type	Value	Description
maps-event-severity-type	string	length: 0..10 pattern: (info|warning|error|critical|^$)	The severity level of the event. Each event has it own severity and you can change it. If the value is empty, the system assigns the severity to the rules. For more information about the system defined severity, refer to the Brocade Fabric OS MAPS User Guide.

maps-generic-action-type

Name	Type	Value	Description
maps-generic-action-type	enumeration	port-fence: The port fencing action. MAPS fences the port to disable the port. snmp-trap: THe SNMP trap action. MAPS sends a SNMP alert to application. raslog: The RASLog action. MAPS logs the message on the console and syslog server. sddq: The Slow-Drain Device Quarantine (SDDQ) action. MAPS quarantines the port to mitigate the congestion. un-quarantine: The un-quarantine action. MAPS un-quarantines the port when the congestion issue is fixed. decommission: The decommission action. MAPS disables the port without traffic interruption. port-toggle: The port toggle action. MAPS toggles the port to fix congestion issue. e-mail: The e-mail action. fms: The FMS action. MAPS notifies the FICON host. vtap-uninstall: The VTAP action. re-balance: The re-balance action. MAPS balances the port group state. sw-marginal: The switch marginal action. MAPS puts the switch into a marginal operating state. The state is a logical state. sw-critical: The switch critical action. MAPS puts the switch into the down operating state. The state is a logical state. sfp-marginal: Represents sfp marginal action. MAPS puts the SFP into the marginal operating state. The state is a logical state. fpin: The Fabric Performance Impact Notifications (FPIN) action. MAPS sends an FPIN to registered end devices. none: No action.	The MAPS actions. You can enable one or more actions at the switch level or per rule.

maps-group-feature-type

Name	Type	Value	Description
maps-group-feature-type	string	length: 0..32 pattern: (port\-name|node\-wwn|^$)	The existing feature name for the group. Valid feature names are port-name and node-wwn. You can create user-defined groups matching the port name or node WWN. If the value is empty, no user feature is assigned to the group.

maps-group-name-type

Name	Type	Value	Description
maps-group-name-type	string	length: 1..32 pattern: [0-9a-zA-Z_]+	The name of the group to be configured. The group name must be unique; it is not case sensitive and can contain up to 32 characters.

maps-group-type-type

Name	Type	Value	Description
maps-group-type-type	enumeration	power-supply: The power supply group type. The power supply group manages the power supplies. The system can have multiple power supplies and can be integrated with a fan. fan: The fan group type. The fan group manages the fans present in the system. The system can have multiple fans can be integrated with the power supplies. fc-port: The FC port group type. The FC port group manages the FC ports. sfp: The FC SFP group type. The SFP group manages the FC SFPs. The system can have different SFPs based on speed or vendor. blade: The blade group type. The blade group manages all blades (such as core, CP, or switch blade). circuit: The circuit group type. The circuit group manages circuits. A circuit is a virtual connection between two extension systems. circuit-qos: The Circuit QOS group type. The Circuit QOS group manages the Circuit QOS traffic. Each circuit can carry multiple QOS traffic. temperature-sensor: The temperature sensor group type. The temperature sensor group manages temperature sensors. flash: The flash memory group type. switch: The switch group type. chassis: The chassis group type. cpu: The CPU group type. wwn: The WWN group type. The WWN group manages the WWN cards. flow: The flow group type. tunnel: The tunnel group type. The tunnel group manages the tunnels. A tunnel is group of circuits. tunnel-qos: The Tunnel QOS group type. The Tunnel QOS group manages each QOS per tunnel. backend-port: The Back-End group type. The Back-End group manages the Back-End ports. ge-port: The Gigabit Ethernet port group type. certificate: The security certificate group type. dp: The data process group type. device-pid: The device PID group type. ethernet-port: The Ethernet port group type.	The type of elements present in the group.

maps-logical-operator-type

Name	Type	Value	Description
maps-logical-operator-type	enumeration	l: The less than logical operator. le: The less than or equal to logical operator. g: The greater than logical operator. ge: The greater than or equal to logical operator. eq: The equal to logical operator. ne: The not equal to logical operator.	The operator to be used for evaluating the condition.

maps-monitoring-system-type

Name	Type	Value	Description
maps-monitoring-system-type	string	length: 2..72 pattern: [0-9a-zA-Z_]+	The monitoring system (MS). MAPS uses a monitoring system to monitor stats, performance, or faulty. Examples: CRC, BLADE_STATE.

maps-policy-name-type

Name	Type	Value	Description
maps-policy-name-type	string	length: 1..32 pattern: [0-9a-zA-Z_]+	The name of the policy to be created. The policy name must be unique; it is case-sensitive and can contain up to 32 characters.

maps-port-monitoring-state

Name	Type	Value	Description
maps-port-monitoring-state	enumeration	offline: The offline state. online: The online state. faulty: The faulty state. marginal: The marginal state. A port can be in a marginal state if the port does not have enough credits to operate. error: The error state. A port can be set in the error state if the port is not in operation due to hardware malfunctioning or security. un-monitored: The un-monitored state.	MAPS monitors each port and based on its operating condition it sets the state.

maps-quiet-time-unit-type

Name	Type	Value	Description
maps-quiet-time-unit-type	enumeration	minute: The QT unit as a minute. hour: The QT unit as an hour. day: The QT unit as a day.	The unit of the quiet time.

maps-rule-name-type

Name	Type	Value	Description
maps-rule-name-type	string	length: 1..72 pattern: [0-9a-zA-Z_]+	The rule name.

maps-rule-type

Name	Type	Value	Description
maps-rule-type	boolean		The rule name. A rule can be a base rule or rule-on-rule (RoR) rule. Base rules monitor the stats or a FRU. RoR rules monitor the base rule.

maps-time-base-type

Name	Type	Value	Description
maps-time-base-type	enumeration	none: The none time base monitors in real time. second: The rule is evaluated every second. minute: The rule is evaluated every minute. five-minute: The rule is evaluated every five minutes. hour: The rule is evaluated every hour. day: The rule is evaluated every day. week: The rule is evaluated every week. 10-second: The rule is evaluated every 10 seconds. IO: The rule is monitored per IO in the hardware.	The time interval between two samples to be compared.

media-alerts-group

Description

A list of warning and alarm levels for media

Parameters

Table 236: media-alerts-group

Name	Type	Description
high-alarm	alert-type	The high alarm for the media.
low-alarm	alert-type	The low alarm for the media.
high-warning	alert-type	The high warning level for the media.
low-warning	alert-type	The low warning level for the media.

media-optical-product-data-group

Description

A list of optical product data for media

Parameters

Table 237: media-optical-product-data-group

Name	Type	Description
part-number	fru:part-number-type	The part number assigned by the organization responsible for producing or manufacturing the media.
serial-number	fru:serial-number-type	The factory's serial number for the blade.
vendor-name	vendor-name-type	The vendor name of the media.
vendor-revision	vendor-revision-type	The vendor revision number of the media.
date-code	date-code-type	The vendor's manufacturing date code.

media-power-type

Name	Type	Value	Description
media-power-type	decimal64	range: 0..9999.99	The power type of the media.

message-id-type

Name	Type	Value	Description
message-id-type	string	length: 7..12 pattern: [0-9A-Z]{2,7}-[0-9]{4,4}	Message ID Type for the raslog.

mibs-name

Name	Type	Value	Description
mibs-name	enumeration	FE-MIB: It indicates FE-MIB. SW-MIB: It indicates SW-MIB. FA-MIB: It indicates FA-MIB. FICON-MIB: It indicates FICON-MIB. HA-MIB: It indicates HA-MIB. FCIP-MIB: It indicates FCIP-MIB. IF-MIB: It indicates IF-MIB. BROCADE-MAPS-MIB: It indicates BROCADE-MAPS-MIB. T11-FC-ZONE-SERVER-MIB: It indicates T11-FC-ZONE-SERVER-MIB.	The list of supported snmp mibs.

module-id-type

Name	Type	Value	Description
module-id-type	string	length: 2..7 pattern: [0-9A-Z]{2,7}	The type of FOS module-id.

monitoring-type-type

Name	Type	Value	Description
monitoring-type-type	enumeration	event-based: Event-based monitoring. poll-based: Poll-based monitoring.	The monitoring type. Typically, all stats monitoring are poll-based and a change in the stats can trigger the rule. Whereas, an event-based monitoring rule is triggered due to a specific event (such as blade out).

month-type

Name	Type	Value	Description
month-type	enumeration	jan: January feb: February mar: March apr: April may: May jun: June jul: July aug: August sep: September oct: October nov: November dec: December	The month of the year.

ms-acl-access-type

Name	Type	Value	Description
ms-acl-access-type	enumeration	no-access: No commands allowed. read-only: Only read commands allowed. read-write: Read and write commands allowed. invalid: Configuration is invalid. Will effectively behave as if access is read-write.	Define the valid access states of the Management Server Access Control List feature.

notification-capabilities-type

Name	Type	Value	Description
notification-capabilities-type	enumeration	agpin: Aggregator Performance Impact Notification (AGPIN) ELS Command	Fabric Notification capabilities.

ntp-clock-server-group

Description

This common model is used for both time server module

Parameters

Table 238: ntp-clock-server-group

Name	Type	Description
server	ts-ntp-type	NTP server IP address or LOCL (for the local server). Configure an NTP server IP address to set the clock server IP address and enable fabric-wide clock synchronization with the specified clock server.
index	int32	NTP symmetric key configured for the NTP server. Configure an NTP key index to be associated with the NTP server for symmetric encryption. Valid indexes are from 1 to 65535. -1 will be returned when index is not in use.

object-identifier-128

Name	Type	Value	Description
object-identifier-128	object-identifier	pattern: \d*(\.\d*){1,127}	This type represents object-identifiers restricted to 128 sub-identifiers. In the value set and its semantics, this type is equivalent to the OBJECT IDENTIFIER type of the SMIv2.

object-identifier

Name	Type	Value	Description
object-identifier	string	pattern: (([0-1](\.[1-3]?[0-9]))|(2\.(0|([1-9]\d*))))(\.(0|([1-9]\d*)))*	The object-identifier type represents administratively assigned names in a registration-hierarchical-name tree. Values of this type are denoted as a sequence of numerical non-negative sub-identifier values. Each sub-identifier value MUST NOT exceed 2^32-1 (4294967295). Sub-identifiers are separated by single dots and without any intermediate whitespace. The ASN.1 standard restricts the value space of the first sub-identifier to 0, 1, or 2. Furthermore, the value space of the second sub-identifier is restricted to the range 0 to 39 if the first sub-identifier is 0 or 1. Finally, the ASN.1 standard requires that an object identifier has always at least two sub-identifiers. The pattern captures these restrictions. Although the number of sub-identifiers is not limited, module designers should realize that there may be implementations that stick with the SMIv2 limit of 128 sub-identifiers. This type is a superset of the SMIv2 OBJECT IDENTIFIER type since it is not restricted to 128 sub-identifiers. Hence, this type SHOULD NOT be used to represent the SMIv2 OBJECT IDENTIFIER type; the object-identifier-128 type SHOULD be used instead.

octet-member-index-type

Name	Type	Value	Description
octet-member-index-type	int8	range: 0..7	Used to identify which member in the octet for a given trunk.

octet-speed-combo-type

Name	Type	Value	Description
octet-speed-combo-type	enumeration	2g-4g-8g-16g-32g-mode: Auto-negotiated or fixed port speeds of 32Gbps, 16 Gbps, 8 Gbps, 4 Gbps, and 2 Gbps 2g-4g-8g-10g-mode: Auto-negotiated or fixed port speeds of 10 Gbps, 8 Gbps, 4 Gbps, and 2 Gbps 10g-16g-mode: Auto-negotiated or fixed port speeds of 16 Gbps and 10 Gbps	Supported modes of octet speed combo for a port. The octet combination default is 2g-4g-8g-16g-32g-mode.

operational-state-type

Name	Type	Value	Description
operational-state-type	string	length: 2..60 pattern: faulty|ok|below minimum|above maximum|absent|unknown|not ok	The current operational state of the Fru. The state 'not ok' displays when the unit is faulty for an unknown reason.

operational-status-type

Name	Type	Value	Description
operational-status-type	enumeration	enabled: The switch is in an online state. disabled: The switch is in an offline state. undefined: The switch is in an undefined state. testing: The switch is being tested.	The operational status of the switch.

operational-type

Name	Type	Value	Description
operational-type	uint16	range: 0..40	Tunnel/Circuit operational status: 0 = NULL, 1 = Offline, 2 = Online, 3 = Online Warning, 4 = Disabled, 5 = Degraded, 6 = Initializing, 7 = Delete Pending, 8 = HA Online, 9 = HA Offline, 10 = HA Ready, 11 = Empty, 12 = In Progress, 13 = MisConfig, 14 = Failover, 15 = Down Pending, 16 = Circuit Disabled/Fenced/Testing, 17 = Internal Error, 18 = IPSec Error, 19 = Network Error, 20 = Authentication Error, 21 = Timeout, 22 = TCP-Timeout, 23 = Remote Close Timeout, 24 = Remote Close, 25 = Rejected, 26 = No Port, 27 = No Route, 28 = DP Offline, 29 = HCL Inprogress, 30 = Internal Error, 31 = Configuration Incomplete, 32 = Circuit Fenced, 33 = Child Delete Complete, 34 = Delete Failure, 35 = Spill Over, 36 = Running, 37 = Testing, 38 = Aborted, 39 = Passed, 40 = Failed

optical-type

Name	Type	Value	Description
optical-type	enumeration	optical-port-type: The remote media connector is an optical type. not-optical: The remote media connector is not an optical type.	Whether the remote media connector is an optical type.

oversubscription-state-type

Name	Type	Value	Description
oversubscription-state-type	enumeration	oversubscription: The oversubscription state. no-oversubscription: The no oversubscription state. monitoring-paused: Monitoring is paused for the member.	The available states on the FPI oversubscription conditions.

part-number-type

Name	Type	Value	Description
part-number-type	string	length: 0..16 pattern: [-~]{0,16}	The part number assigned by the organization responsible for producing or manufacturing the PhysicalElement.

password-cfg-operation-type

Name	Type	Value	Description
password-cfg-operation-type	enumeration	hash-config: Sets the hash type. Valid values are md5, sha256, or sha512. default: Configures all password policies to their default value. delete-all: Removes the password expiration policies of all users.	Create/Delete Password policy configuration

percentage-type

Name	Type	Value	Description
percentage-type	string	length: 1..6 pattern: ([0-9][0-9].[0-9][0-9]|[0-9].[0-9][0-9]|100.00)	Percentage string up to two decimal places.

performance-group-id-type

Name	Type	Value	Description
performance-group-id-type	string	length: 4..64 pattern: ([a-zA-Z0-9][0-9a-zA-Z-._]{3,63})	The Performance Group name to identify PG

phys-address

Name	Type	Value	Description
phys-address	string	pattern: ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2})*)?	Represents media- or physical-level addresses represented as a sequence octets, each octet represented by two hexadecimal numbers. Octets are separated by colons. The canonical representation uses lowercase characters. In the value set and its semantics, this type is equivalent to the PhysAddress textual convention of the SMIv2.

physical-interface-type

Name	Type	Value	Description
physical-interface-type	enumeration	eth0: Physical ethernet interface eth0 eth3: Physical ethernet interface eth3.	Name of the phsyical management interface. For fixed-port switch, valid name is eth0. For chassis switch, valid names are eth0 and eth3.

pki-key-pair

Name	Type	Value	Description
pki-key-pair	string	length: 1..32 pattern: / modifier: invert-match	For 'pki' profile, this field specifies the local key pair name to use for IKE authentication. This operand is required with PKI profile.

port-address-binding-mode

Name	Type	Value	Description
port-address-binding-mode	enumeration	8-bit: 8-bit binding mode for the port 9-bit: 8-bit binding mode for the port 10-bit: 10-bit binding mode for the port none: none of the above binding modes	It enumerates port-address-binding mode that is supported

port-decommission-operation-type

Name	Type	Value	Description
port-decommission-operation-type	enumeration	port: Decommission a port qsfp-port: Decommission a QSFP module	Enumerates the types of port decommissioning operations that are available.

port-group-state-type

Name	Type	Value	Description
port-group-state-type	enumeration	unknown: The unknown state. balanced: The balanced state of the port group. imbalanced: The imbalanced state of the port group. balance-failed: The balanced failed state of the port group. It occurs when MAPS is unable to bring the group state back to a balanced state.	The port group states.

port-number-type

Name	Type	Value	Description
port-number-type	user-port-number-type	range: 0..3400	Count of ports or port numbers.

port-number

Name	Type	Value	Description
port-number	uint16	range: 0..65535	The port-number type represents a 16-bit port number of an Internet transport-layer protocol such as UDP, TCP, DCCP, or SCTP. Port numbers are assigned by IANA. A current list of all assignments is available from <http://www.iana.org/>. Note that the port number value zero is reserved by IANA. In situations where the value zero does not make sense, it can be excluded by subtyping the port-number type. In the value set and its semantics, this type is equivalent to the InetPortNumber textual convention of the SMIv2.

port-operation-type

Name	Type	Value	Description
port-operation-type	enumeration	port-address-bind: Bind operation for port address. port-address-unbind: Unbind operation for port address. port-reset-statistics: reset the statistical-counters on the given port extension-ge-port-reset-statistics: Reset the Extension GE port statistics. This operation is valid only for Extension Platforms.	It enumerates port operations supported.

port-scn-type

Name	Type	Value	Description
port-scn-type	enumeration	online: The port's last scn received is online. offline: The port's last scn received is offline. testing: The port's last scn received is testing. faulty: The port's last scn received is faulty. e-port: The port's last scn received is E-Port. t-port: The port's last scn received is T-Port. f-port: The port's last scn received is F-Port. segmented: The port's last scn received is segmented. vf-port: The port's last scn received is VF-Port. unknown: The port's last scn received is unknown.	Supported port-scn types.

port-type-string-type

Name	Type	Value	Description
port-type-string-type	enumeration	e-port: Port is an E-port. g-port: Port is a G-port. universal-port: If port is not configured for any type. f-port: Port is a F-port. l-port: Port is a Logical port. fcoe-port: Port is a FCoE-port. ex-port: Port is a EX-port. d-port: Port is Diagnostic port. sim-port: Port is SIM-port. af-port: Port is AF-port. Present only on AMP switches. ae-port: Port is an AE-port. ve-port: Port is a VE-port. ethernet-port: Port is a Ethernet Flex port. flex-port: Port is a Flex-port. n-port: Port is N-port. Applicable for AG mode. mirror-port: Port is Mirror-port. encryption-support-port: Port is an Encryption Support Port. A front end port reserved for Encryption block to work. loopback-port: Port is a Loopback port. unknown-port: If Port is online and does not belong any known port type. icl-port: Port is a ICL port. fc-lag-port: Port is a FC-LAG member port.	Supported Fibre Channel port types:

port-type-type

Name	Type	Value	Description
port-type-type	uint16	range: 0|7|10|11|15|16|17|19|20|21|22|23|25|26|29|30|31|32|33|32768	Fibre Channel port types: 0 : Unknown 7 : E Port 10 : G Port 11 : U Port 15 : F Port 16 : L Port 17 : FCoE Port 19 : EX Port 20 : D Port 21 : SIM Port 22 : AF Port 23 : AE Port 25 : VE Port 26 : Ethernet Flex Port 29 : Flex Port 30 : N Port 31 : Mirror Port 32 : ICL Port 33 : FC-LAG Port 32768 : LB Port Deprecated. Please use port-type-string-type in stead.

port-type

Name	Type	Value	Description
port-type	enumeration	fc-port: A logical grouping for FC ports. be-port: A logical grouping for Back-End ports. ge-port: A logical grouping for Gigabit Ethernet ports.	A logical grouping for different port counters.

portchannel-interface-name-type

Name	Type	Value	Description
portchannel-interface-name-type	string	length: 0..31 pattern: [0-9a-zA-Z_-]*	The name of a Portchannel interface.

portchannel-speed-type

Name	Type	Value	Description
portchannel-speed-type	uint64	range: 0|1000000000|10000000000| 25000000000|40000000000	The current speed of a portchannel. 0 : Auto negotiated. 1000000000 : 1 GbE 10000000000 : 10 GbE 25000000000 : 25 GbE 40000000000 : 40 GbE By default, the speed of the portchannel is 0 if the port-channel does not have any member port. For 'auto' negotiated speed, the speed of the portchannel will be update to the speed of the portchannel member interface. Note: The speed of all members of a portchannel must match.

portchannel-type

Name	Type	Value	Description
portchannel-type	enumeration	static: The static portchannel type. dynamic: The dynamic portchannel type.	The portchannel can be of either static or dynamic type

portchannel_platform

Indicates that Port Channels are supported on this platform.

This feature requires the switch to be in Native mode.

power-detail-type

Name	Type	Value	Description
power-detail-type	int32	range: -1000..-1	The power details such as the maximum power consumption allocated or consumed by the fru.

power-state-type

Name	Type	Value	Description
power-state-type	enumeration	on: to power on the slot. off: to power off the slot	It indicates the power state of the fru.

preshared-key

Name	Type	Value	Description
preshared-key	string	length: 16..64 pattern: [a-zA-Z0-9~@%_+:\[\]][a-zA-Z0-9~@%_+:\[\]\-]{15,63}	For 'preshared' profile, this field specifies the preshared-key to be used for authentication. This operand is required with shared-key authentication.

privacy-protocol-type

Name	Type	Value	Description
privacy-protocol-type	enumeration	des: The privacy protocol DES is unsafe and not recommended. Using AES is recommended. aes128: Indicates the privacy protocol as AES128. aes256: Indicates the privacy protocol as AES256. nopriv: Indicates the privacy protocol as none.	It indicates the SNMPV3 privacy protocols

product-name-type

Name	Type	Value	Description
product-name-type	string	length: 1..255 pattern: [-~]{1,255}	The typedef defines a common Brocade Switch, Chassis or Blade product names known to customer. Example Extension Platforms : 7810/7840/ SX6(blade)

protocol-speed-type

Name	Type	Value	Description
protocol-speed-type	enumeration	not-established: Speed and/or protocol information not available. 1-gfc: 1 Gigabit Fibre Channel 2-gfc: 2 Gigabit Fibre Channel 4-gfc: 4 Gigabit Fibre Channel 8-gfc: 8 Gigabit Fibre Channel 10-gfc: 10 Gigabit Fibre Channel 12-gfc: 12 Gigabit Fibre Channel 16-gfc: 16 Gigabit Fibre Channel 20-gfc: 20 Gigabit Fibre Channel 24-gfc: 24 Gigabit Fibre Channel 28-gfc: 28 Gigabit Fibre Channel 32-gfc: 32 Gigabit Fibre Channel 40-gfc: 40 Gigabit Fibre Channel 48-gfc: 48 Gigabit Fibre Channel 53-gfc: 53 Gigabit Fibre Channel 56-gfc: 56 Gigabit Fibre Channel 64-gfc: 64 Gigabit Fibre Channel 80-gfc: 80 Gigabit Fibre Channel 96-gfc: 96 Gigabit Fibre Channel 112-gfc: 112 Gigabit Fibre Channel 128-gfc: 128 Gigabit Fibre Channel 160-gfc: 160 Gigabit Fibre Channel 192-gfc: 192 Gigabit Fibre Channel 224-gfc: 224 Gigabit Fibre Channel 256-gfc: 256 Gigabit Fibre Channel 320-gfc: 320 Gigabit Fibre Channel 384-gfc: 384 Gigabit Fibre Channel 448-gfc: 448 Gigabit Fibre Channel 512-gfc: 512 Gigabit Fibre Channel 640-gfc: 640 Gigabit Fibre Channel 768-gfc: 768 Gigabit Fibre Channel 896-gfc: 896 Gigabit Fibre Channel 1024-gfc: 1024 Gigabit Fibre Channel 1280-gfc: 1280 Gigabit Fibre Channel 1536-gfc: 1536 Gigabit Fibre Channel 1792-gfc: 1792 Gigabit Fibre Channel 2048-gfc: 2048 Gigabit Fibre Channel 1-ge: 1 Gigabit Ethernet 10-ge: 10 Gigabit Ethernet 25-ge: 25 Gigabit Ethernet 40-ge: 40 Gigabit Ethernet 50-ge: 50 Gigabit Ethernet 100-ge: 100 Gigabit Ethernet 400-ge: 400 Gigabit Ethernet bit-0: Unrecognized speed classification. See FC-GS FDMI.	Specifies numeric link speed (or bandwidth) with respect to the Ethernet (Layer-2) or Fibre Channel (Physical Interface) protocol. This value applies to individual ports, and Link Aggregation Groups such as a Fibre Channel Trunk, Fibre Channel LAG, or Port Channel.
		bit-1: Unrecognized speed classification. See FC-GS FDMI. bit-2: Unrecognized speed classification. See FC-GS FDMI. bit-3: Unrecognized speed	

protocol

Name	Type	Value	Description
protocol	enumeration	scp: The Secure Copy Protocol (from or to a remote host. ftp: The File Transfer Protocol. sftp: The Secure File Transfer Protocol. http: The HyperText Transfer Protocol.	The protocol (FTP, SCP, SFTP, or HTTP).

rbac-class-permission-type

Name	Type	Value	Description
rbac-class-permission-type	string	length: 1..67 pattern: [a-zA-Z]{1,64}=(OM|O)	Class name with '=' separated class permission. Example : LocalUserEnvironment=O

rbac-class-type

Name	Type	Value	Description
rbac-class-type	string	length: 1..64 pattern: [a-zA-Z]+	The role based access control class name

rbac-context-type

Name	Type	Value	Description
rbac-context-type	enumeration	chassis: chassis context. vf: virtual fabric context. both: In virtual fabric context and chassis context disallowed: Disallowed in non virtual fabric context	Specifies context permission.

rbac-permission-type

Name	Type	Value	Description
rbac-permission-type	enumeration	N: no permission. O: observe permission. M: modify only permission. OM: observe-modify permission.	Specifies RBAC permission type.

rbac-role-type

Name	Type	Value	Description
rbac-role-type	enumeration	system-defined: Default role user-defined: User defined role	Specifies the role type. system-defined or user-defined.

reboot-types

Name	Type	Value	Description
reboot-types	enumeration	reboot: Performs a cold reboot (power off/restart) of the control processor. This operation may be disruptive. When you reboot a switch connected to a fabric, all traffic to and from that switch stops. All Fibre Channel ports on that switch including E_Ports become inactive until the switch comes back online. The behavior of this command depends on the platform, refer to the command reference manual for additional detail about the reboot command. sysshutdown: Provides a graceful shutdown to protect the switch file systems. On standalone platforms, use sysShutDown to shut down the switch operating system. On enterprise-class platforms, when sysShutDown is called on the active control processor (CP), the command shuts down the active CP, standby CP, and any AP blades. After executing this command, manually power off the system. To reboot the system, manually turn the power switch on. Note : This command is not supported on the standby CP. See Using Fabric OS Commands and Command Availability in the command reference manual for details.	Specifies how to reboot the system.

reliable-type

Name	Type	Value	Description
reliable-type	enumeration	unreliable: N_Port link is Unreliable. reliable: N_Port link is Reliable. not-applicable: N_Port link status is Not applicable.	The reliable status of the N_Port.

result-type

Name	Type	Value	Description
result-type	string	length: 1..64 pattern: NOT STARTED|PASSED|SKIPPED|FAILED|IN PROGRESS|RESPONDER|STOPPED|WAITING FOR FABRIC STABLE|UNKNOWN	Result is the completion status of the test. Valid values are: NOT STARTED PASSED SKIPPED FAILED IN PROGRESS RESPONDER STOPPED UNKNOWN

role-name-type

Name	Type	Value	Description
role-name-type	string	length: 4..16 pattern: [a-z]{4,16}	Specifies the account's role name.

sec-crypto-cfg-actions-type

Name	Type	Value	Description
sec-crypto-cfg-actions-type	enumeration	apply: Activate a default or user-defined template file. verify: Verifies the running configuration against a required configuration specified in the template file. If this operation returns '204 No Content', the verify was successful. import: Imports a template file from a specified external host. The template name should not be same as default template names when importing the file. export: Exports a template file to the specified external host. Template name should be that of a default or user defined template that already exists. Following are the default templates, o default_generic o default_strong o default_fips o default_cc	o apply - Enforce default or user-defined template file. o verify - Verifies the running configuration against a required configuration specified in the template file. o import - Imports a template file from a specified external host. o export - Exports a template file to the specified external host.

sec-crypto-cfg-default-template-name-type

Name	Type	Value	Description
sec-crypto-cfg-default-template-name-type	enumeration	default_generic: Generic Template for Security Crypto Configuration. default_strong: Template for High Security Crypto Configuration. default_fips: FIPS Template for Security Crypto Configuration. default_cc: Common Criteria (CC) Template for Security Crypto Configuration.	Specify the names of default templates.

sec-crypto-cfg-file-transfer-protocol-type

Name	Type	Value	Description
sec-crypto-cfg-file-transfer-protocol-type	enumeration	scp: scp type sftp: sftp type ftp: ftp type	Specify SCP or SFTP or FTP for file transfer.

sec-crypto-cfg-template-name-type

Name	Type	Value	Description
sec-crypto-cfg-template-name-type	string	length: 1..256 pattern: [a-zA-Z0-9_]{1,256}	Specify name of the template. Default template names such as, o default_generic o default_cc o default_strong o default_fips cannot be used for user defined template addition or deletion.

sec-crypto-cfg-tls-cipher-type

Name	Type	Value	Description
sec-crypto-cfg-tls-cipher-type	string	length: 1..64 pattern: [-+!0-9]{0,1}[a-zA-Z0-9]{2,64}	Cipher algorithms are valid for HTTPS, RADIUS, LDAP, and SYSLOG.

seccertmgmt-action-type

Name	Type	Value	Description
seccertmgmt-action-type	enumeration	import: Import certificates from the server or to download a certificate issued by a CA after sending the CSR to the CA.	certificate management action.

seccertmgmt-hash-type

Name	Type	Value	Description
seccertmgmt-hash-type	enumeration	sha1: Secure Hash Algorithm that generates a 160-bit (20-byte) hash value. sha256: Secure Hash Algorithm that generates a unique, fixed size 256-bit (32-byte) hash. sha384: Secure Hash Algorithm that generates a unique, fixed size 384-bit (48-byte) hash. Not supported with DSA algorithm key type. Not supported by FCAP. sha512: Secure Hash Algorithm that uses 64 byte words. Not supported with DSA algorithm key type. Not supported by FCAP.	Specifies the hash type.

seccertmgmt-operation-type

Name	Type	Value	Description
seccertmgmt-operation-type	enumeration	import: Import certificates from the server or to download a certificate issued by a CA after sending the CSR to the CA. export: Export certificate to a host and a CSR to server to the Certification Authority (CA) that issues the certificate.	certificate management operations.

seccertmgmt-protocol-type

Name	Type	Value	Description
seccertmgmt-protocol-type	enumeration	scp: secure copy from/to remote host ftp: File Transfer Protocol	Specifies the protocol as either FTP or SCP.

seccrypto-import-export-param-group

Description

This model is used for importing and exporting templates

to/from host.

Parameters

Table 239: seccrypto-import-export-param-group

Name	Type	Description
remote-user-name	string	Specifies the user name for the host.
remote-host-ip	inet:host	Specifies the remote host ip address.
remote-user-password	string	Password for the remote user. Password must be encoded with Base64 encoding scheme
remote-directory	string	Specifies the remote directory fully qualified path name(FQPN). If action is import, remote directory must be fully qualified file name. If the action is export, remote directory can be fully qualified file name (to export template with a name of user's choice) or directory name (to export template with the name same as that of template on switch).
file-transfer-protocol-type	sec-crypto-cfg-file-transfer-protocol-type	Specifies the protocol as either SCP, SFTP, or FTP for file transfer

security-compliance-action-type

Name	Type	Value	Description
security-compliance-action-type	enumeration	zeroize: Zeroize compliance related configuration data. self-test-all: Run all Self Tests. self-test-default: Run default Self Tests. self-test-dh: Run DH Self Tests. self-test-drbg: Run DRBG Self Tests. self-test-aes: Run AES Self Tests. self-test-aes-gcm: Run AES-GCM Self Tests. self-test-rsa: Run RSA Self Tests. self-test-rsa2048-sha256: Run RSA2048_SHA256 Self Tests. self-test-des: Run DES Self Tests. self-test-dsa: Run DSA Self Tests. self-test-sha1: Run SHA1 Self Tests. self-test-sha256: Run SHA256 Self Tests. self-test-sha384: Run SHA384 Self Tests. self-test-sha512: Run SHA512 Self Tests. self-test-hmac-sha1: Run HMAC-SHA1 Self Tests. self-test-hmac-sha224: Run HMAC-SHA224 Self Tests. self-test-hmac-sha256: Run HMAC-SHA256 Self Tests. self-test-hmac-sha384: Run HMAC-SHA384 Self Tests. self-test-hmac-sha512: Run HMAC-SHA512 Self Tests. self-test-sha3: Run SHA3 Self Tests. self-test-ecdsa: Run ECDSA Self Tests. self-test-ecdh: Run ECDH Self Tests. crypto-integrity-test: Run crypto integrity test.	Specifies compliance configuration to apply.

security-info-group

Description

Information about a switch or device

Parameters

Table 240: security-info-group

Name	Type	Description
switch-wwn	fibrechannel:wwn-type	The switch-wwn member contains an 8 byte value that identifies the switch that contains the E_Port.
domain-id	fibrechannel:domain-id-type	The highest level in a three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
switch-name	string	A printable ASCII name for the switch.

security-input-type

Name	Type	Value	Description
security-input-type	union	fibrechannel:wwn-type fibrechannel:domain-id-type string	Input for creating security policy operations and authentication secret. WWN or domain ID or switch name. A wildcard (*) may be specified to include all switches in the fabric. Not Applicable for GET instance.

security-level-string-type

Name	Type	Value	Description
security-level-string-type	enumeration	no-security: Indicates the security level as NO Security authentication-only: Indicates the security level as Authentication only authentication-and-privacy: Indicates the security level as Auth and Priv. no-access: Indicates the security level as No access.	It indicates the SNMP security level in string format

security-level

Name	Type	Value	Description
security-level	uint16	range: 0..3	The security access level for SNMP get / set operation. The range values represent the security level as follows. 0 : No Security 1 : Authentication only 2 : Authentication and privacy 3 : No Access deprecated: Please use 'security-level-string-type' instead.

security-policy-action-type

Name	Type	Value	Description
security-policy-action-type	enumeration	create: Creates a security policy. delete: Deletes a security policy. add: Adds member(s) to an existing security policy. remove: Removes member(s) to an existing security policy. fcs-move: Moves a member in the FCS policy to the specified position. activate: Changes all defined policies into activated state.	Indicates the action that must be taken on security policies.

security-policy-default-name-type

Name	Type	Value	Description
security-policy-default-name-type	enumeration	SCC_POLICY: Switch Connection Control policy. FCS_POLICY: Fabric Configuration Server policy. ALL_DCC_POLICY: Deletes all DCC policies from the defined policy list. Applicable only for delete instance. ALL_STALE_DCC_POLICY: Deletes all stale DCC policies from the defined policy list. Applicable only for delete instance.	Indicates the policy name for security policies.

security-policy-distribute-permission-type

Name	Type	Value	Description
security-policy-distribute-permission-type	enumeration	accept: Configures switch to accept distributions of the specified policy. reject: Configures switch to reject distributions of the specified policy. However, a database cannot be rejected if it is specified in the fabric-wide consistency policy.	Switch can be locally configured to allow or reject a security policy.

security-policy-name-type

Name	Type	Value	Description
security-policy-name-type	union	security-policy-default-name-type string	Device Connection Control policy. Name has the common prefix DCC_POLICY_ followed by a string of user-defined characters. These characters do not have to be capitalized.

security-policy-state-type

Name	Type	Value	Description
security-policy-state-type	enumeration	active: The current status of the policy is active. defined: The current status of the policy is defined.	The current status of the policy.

security-violation-statistics-type

Name	Type	Value	Description
security-violation-statistics -type	enumeration	telnet-policy: Telnet policy violation statistics. http-policy: HTTP policy violation statistics. scc-policy: SCC policy violation statistics. dcc-policy: DCC policy violation statistics. login: Login violation statistics. authentication-fail: Authentication failure statistics. ts-out-sync: Time server out of sync statistics. illegal-command: Illegal command execution statistics. all: Include all the above mentioned statistics.	Name of the security violation statistics.

segmentation-reason-type

Name	Type	Value	Description
segmentation-reason-type	enumeration	loopback: The port has a loopback cable. incompatible: The switches are incompatible. overlap: The domains are overlapping. zoning: A zoning conflict. e-port-segment: An E_Port segmentation. no-license: No license. disabled-e-port: The port is disabled to be an E_Port. platform-db: An incompatible platform database. security-incompatibility: A security incompatibility. security-violation: A security violation. ecp-error: The Exchange Credit Parameter is rejected or exceeded retries. duplicate-wwn: A duplicate WWN in the fabric. e-port-isolated: The E_Port is isolated. vf-ad-conflict: A Virtual Fabric admin domain conflict. esc-detected-conflict: An Exchange Switch Capabilities detected conflict. encryption-conflict: An Encryption conflict. compression-conflict: A compression conflict. encryption-or-compression-bandwidth-unavailability: An Encryption or Compression bandwidth unavailability. defzone-conflict: A defzone conflict. alias-peer-zone-conflict: An alias peer zone conflict. enhanced-zone-object-naming-conflict: An enhanced zone object naming conflict. chassis-zone-db-size-exceeded: A chassis zone DB size conflict. fabric-zone-db-size-exceeded: A fabric zone DB size conflict. zone-merge-timed-out: The zone merge timed out. zone-merge-internal-error: A zone merge internal error. ad-configuration-not-supported: The admin domain configuration not supported. encrypted-trunk-port-limit-exceeded: The encrypted trunk port limit exceeded. encryption-trunk-config-mismatch: An encryption trunking configuration	The reasons that caused fabric segmentations.

sensor-category

Name	Type	Value	Description
sensor-category	enumeration	temperature: Indicates type temperature sensor of the fru.	It indicates type of fru sensor.

serial-number-type

Name	Type	Value	Description
serial-number-type	string	length: 0..16 pattern: [-~]{0,16}	The serial-number resource contains a Printable ASCII String that specifies the serial number of the chassis

server-parameters

Description

The details of the remote server from which the license certificate is copied onto a local switch.

Parameters

Table 241: server-parameters

Name	Type	Description
host	inet:host	The IP address or host name of the remote server.
user-name	string	The user name of the remote server that is used to copy the license certificates.
password	string	The password for the remote server used to copy the license certificate. The password must be base64 encoded. Refer to RFC 3414.
remote-directory	string	The file path of the remote server from which the license certificate is to be transferred.
protocol	operation-types:protocol	The transport protocol.
port	uint32	The user-defined port number for scp (Secure Copy Protocol) and sftp (Secure FTP).
remote-file	string	The remote file name is used to export the license certificate. This leaf is required only when you want to export the license certificate into a specified remote filename.

severity-level-type

Name	Type	Value	Description
severity-level-type	string	length: 4..8 pattern: info|warning|error|critical	The type of severity levels.

severity-level

Name	Type	Value	Description
severity-level	enumeration	none: This severity is set none of the log messages are displayed. critical: if this severity is set, the critical and below level log messages are displayed. error: If this severity is set, the error and below level log messages are displayed. warning: If this severity is set, the warning and below level log messages are displayed. informational: If this severity is set, the informational and below level log messages are displayed. debug: If this severity is set, the debug and below level log messages are displayed.	The severity level of event notifications.

show-status-group

Description

The group of leafs associated with the operation.

Parameters

Table 242: show-status-group

Name	Type	Description
show-status	Container	The RPC operation status based on the specified message-id.
- message-id	message-id-type	The random message-id required to retrieve the associated operation status.
- operation	string	The operation name (such as supportsave). The show-status is common across all the operations supported in the yang RPC feature. It identifies the exact operation status when the application queries the status based on the message-id.

Name	Type	Description
- status	string	The exact state of the REST operation (supportSave, firmware download, and configuration download). When an operation is triggered, the 'queued' status displays. While the operation is running the 'in-progress' status displays until all input operations are complete in the backend modules. When the input operation is complete, the 'done' status displays. After the completion status, the 'delivered' status displays until it is purged from the queue. If any operational-level errors or application-level errors occur, the system returns the standard error structure. When the input operation is removed from the queue, the 'purge' status displays.
- application-name	string	The name of the application which is used to trigger the operation. This is parsed from the HTTP USER_AGENT standard header.
- percentage-complete	uint8	The operation completion in a percentage.
- firmwaredownload	Container	A list of firmware download status messages or the EULA.
-- message	string	The status messages, depicting all stages of the firmware download progress.
-- eula-text	string	The contents of the latest EULA (Brocade End User License Agreement).
- configdownload	Container	A list of configuration download status messages or the EULA.
-- message	string	The status messages of a configuration download.

signal-descriptor-type

Name	Type	Value	Description
signal-descriptor-type	enumeration	none: There is no optional signal. loss-of-signal: The loss of signal. loss-of-signal-inverted: The loss of signal is inverted. tx-fault: A fault in the transmitter signal. tx-disable: The transmitter signal is disabled.	The media optional transceiver signals.

slot-configuration-action-type

Name	Type	Value	Description
slot-configuration-action-type	enumeration	default: Set ALL ports of given slot number to default port configuration, regardless of which Logical Switch the ports are being present. power: To power-off or power-on the slot	The action type field

slot-port-name-type

Name	Type	Value	Description
slot-port-name-type	string	length: 0..32 pattern: (([0-9]|1[0-9])/([1][0-9][0-9][0-9]|[1-9][0-9][0-9]|[1-9][0-9]|[0-9]))?	Typically the name of the interface. The slot and port number of the port (slot/port).

speed-type

Name	Type	Value	Description
speed-type	uint64	range: 1000000000|10000000000|25000000000|40000000000|100000000000	The bandwidth of the Port. 1000000000 : 1 GbE 10000000000 : 10 GbE 25000000000 : 25 GbE 40000000000 : 40 GbE 100000000000: 100 GbE

sshutil-algorithm-type

Name	Type	Value	Description
sshutil-algorithm-type	enumeration	rsa: RSA type dsa: DSA type ecdsa: ECDSA type	key algorithm type

sshutil-hash-type

Name	Type	Value	Description
sshutil-hash-type	enumeration	md5: md5 hash type. sha1: sha1 hash type. sha256: sha256 hash type. sha384: sha384 hash type. sha512: sha512 hash type.	Sets the hash type.

sshutil-key-type

Name	Type	Value	Description
sshutil-key-type	enumeration	public-private-key: Generate specified public-private key pair. host-key: Generate ssh host key.	sshutil key generation type

sshutil-operation-type

Name	Type	Value	Description
sshutil-operation-type	enumeration	import: Import public key for a specified user. Imported public keys are used for passwordless incoming SSH connections. export: Export public key. Exported public keys are used for passwordless outgoing SSH connections.	Operations

ssp-state-type

Name	Type	Value	Description
ssp-state-type	enumeration	unknown: The unknown state. down: The down (critical) state - some service is already impacted. marginal: The marginal state - if any FRU is in a marginal state, fix it or replace it with a new one. healthy: The healthy state - normal operating condition.	The switch status policy report state. The state gives the summary health of the switch and individual components. A marginal state means if the corrective action is not taken service may be impacted. A down state means service is already impacted. The SSP state can be set based on the rule having sw_marginal or sw_down actions. Example: for FAN state (BAD_FAN>= 1) sw_marginal

state-type

Name	Type	Value	Description
state-type	enumeration	inserted: Indicates fru is inserted. removed: Indicates fru is removed. invalid: Indicates the invalid state of the fru.	It indicates the fru state.

stream-type

Name	Type	Value	Description
stream-type	decimal64	range: 0..9999.9	The upstream and downstream signal loss details of the media.

switch-addressing-mode-type

Name	Type	Value	Description
switch-addressing-mode-type	enumeration	dynamic-address-mode: The switch addressing mode is dynamic. fixed-address-mode: The switch addressing mode is fixed. 10-bit-address-mode: The switch addressing mode is 10-bit.	The switch address mode type.

tcl-action-type

Name	Type	Value	Description
tcl-action-type	enumeration	allow: The action 'allow' is to allow a LAN traffic based on the traffic-control-list input filters and should not be dropped and sent over the extension tunnel using a specific priority qos as per target details. deny: The action 'deny' is to deny the LAN traffic based on the traffic-control-list input filters and drop them at the ingress. deny-dp: The 'deny-dp' is to deny a LAN ingress traffic to be dropped only when its sent to a specific DP.	The different action support for TCL configurations.

tcp-operation-mode

Name	Type	Value	Description
tcp-operation-mode	enumeration	slow-start: The TCP connection is operating in slow-starts. Pease refer RFC 2001 for more details. congestion-avoidance: The TCP connection is in congestion avoidance algorithm. Pease refer RFC 2001 for more details. fast-retransmits: The TCP connection is in fast-retransmits. Pease refer RFC 2001 for more details. fast-recovery: The TCP connection is in fast-recovery. Pease refer RFC 2001 for more details.	The different action support for TCL configurations.

temp-alarm-type

Name	Type	Value	Description
temp-alarm-type	int32	range: -32767..32767	The high or low temperature alarm type for the media.

temperature-type

Name	Type	Value	Description
temperature-type	int16	range: -128..1000	The temperature of the media.

threshold-value-type

Name	Type	Value	Description
threshold-value-type	string	length: 1..32 pattern: [0-9a-zA-Z,_,\-,.]+	The threshold to be used in the condition.

time-24hr-extended-type

Name	Type	Value	Description
time-24hr-extended-type	string	length: 0|8 pattern: ^$|[0-2][0-9]:[0-5][0-9]:[0-5][0-9]	Specifies time. Time is in 24 hour clock format HH:MM:SS HH : Hours [00 to 23], MM : Minutes [00 to 59], SS : Seconds [00 to 59].

time-24hr-type

Name	Type	Value	Description
time-24hr-type	string	length: 0..5 pattern: ([0-2][0-9]:[0-5][0-9])?	Specifies time. Time is in 24 hour clock format HH:MM HH : Hours [00 to 23], MM : Minutes [00 to 59].

time-detail-type-in-hours

Name	Type	Value	Description
time-detail-type-in-hours	uint32	range: 0..240000	The time details such as the number of hours the FRU has been powered on or was last powered on.

time-detail-type

Name	Type	Value	Description
time-detail-type	uint32	range: 0..10000	The time details such as the number of days the FRU has been powered on or was last powered on.

time-generated-type

Name	Type	Value	Description
time-generated-type	uint32	range: 1..2147483646	The time at which the statistics were queried. This value specified the time since epoch in seconds.

timestamp

Name	Type	Value	Description
timestamp	yang:timeticks		The timestamp type represents the value of an associated timeticks schema node at which a specific occurrence happened. The specific occurrence must be defined in the description of any schema node defined using this type. When the specific occurrence occurred prior to the last time the associated timeticks attribute was zero, then the timestamp value is zero. Note that this requires all timestamp values to be reset to zero when the value of the associated timeticks attribute reaches 497+ days and wraps around to zero. The associated timeticks schema node must be specified in the description of any schema node using this type. In the value set and its semantics, this type is equivalent to the TimeStamp textual convention of the SMIv2.

timeticks

Name	Type	Value	Description
timeticks	uint32		The timeticks type represents a non-negative integer that represents the time, modulo 2^32 (4294967296 decimal), in hundredths of a second between two epochs. When a schema node is defined that uses this type, the description of the schema node identifies both of the reference epochs. In the value set and its semantics, this type is equivalent to the TimeTicks type of the SMIv2.

tls-protocol-type

Name	Type	Value	Description
tls-protocol-type	string	length: 1..64 pattern: any|TLSv1.3|TLSv1.2| TLSv1TLSv1-TLSv1.2TLSv1- TLSv1.3TLSv1.2-TLSv1.3TLSv1- TLSv1TLSv1.2-TLSv1.2TLSv1.3- TLSv1.3	TLS protocol.TLSv1.3 is supported only for https application

topology-report-error-type

Name	Type	Value	Description
topology-report-error-type	enumeration	unreachable-domain: A fabric domain that is unreachable. logical-isl-restriction: A fabric domain that is only reachable via a path that has a Logical ISL restriction. non-routing-domain-restriction: A fabric domain that is only reachable via a path that tries to pass through a non-routing domain.	The type of topology errors that can be reported by FSPF. The affected domain are in the fabric build domain list, but no valid paths were identified by FSPF for routing traffic to the domain.

transmission-types

Name	Type	Value	Description
transmission-types	enumeration	id: An unknown type. longwave: A longwave laser. shortwave: A shortwave laser. led: A light emitting diode. copper: A copper type.	The media transmission type.

traps-name

Name	Type	Value	Description
traps-name	enumeration	swFCPortScn: It indicates swFCPortScn trap. swEventTrap: It indicates swEventTrap trap. swIPv6ChangeTrap: It indicates swIPv6ChangeTrap. swPmgrEventTrap: It indicates swPmgrEventTrap. swFabricReconfigTrap: It indicates swFabricReconfigTrap. swFabricSegmentTrap: It indicates swFabricSegmentTrap. swExtTrap: It indicates swExtTrap. swStateChangeTrap: It indicates swStateChangeTrap. swPortMoveTrap: It indicates swPortMoveTrap. swBrcdGenericTrap: It indicates swBrcdGenericTrap. swDeviceStatusTrap: It indicates swDeviceStatusTrap. swZoneConfigChangeTrap: It indicates swZoneConfigChangeTrap. connUnitStatusChange: It indicates connUnitStatusChange. connUnitEventTrap: It indicates connUnitEventTrap. connUnitPortStatusChange: It indicates connUnitPortStatusChange trap. linkRNIDDeviceRegistration: It indicates linkRNIDDeviceRegistration trap. linkRNIDDeviceDeRegistration: It indicates linkRNIDDeviceDeRegistration trap. linkLIRRListenerAdded: It indicates linkLIRRListenerAdded trap. linkLIRRListenerRemoved: It indicates linkLIRRListenerRemoved trap. linkRLIRFailureIncident: It indicates linkRLIRFailureIncident trap. fruStatusChanged: It indicates fruStatusChanged trap. cpStatusChanged: It indicates cpStatusChanged trap. fruHistoryTrap: It indicates fruHistoryTrap. linkDown: It indicates linkDown trap. linkUp: It indicates linkUp trap. mapsTrapAM: It indicates mapsTrapAM trap. mapsQuietTimeExpirationTrap: It indicates mapsQuietTimeExpirationTrap. t11ZsRequestRejectNotify: It indicates t11ZsRequestRejectNotify trap.	The list of supported SNMP traps.

ts-authspec-type

Name	Type	Value	Description
ts-authspec-type	enumeration	noauth: No encryption. symmetric: Enable symmetric encryption.	NTP authspec modes.

ts-key-type

Name	Type	Value	Description
ts-key-type	enumeration	MD5: HMAC with MD5 key type. HMAC-SHA1: HMAC with SHA1 key type. HMAC-SHA256: HMAC with SHA256 key type. CMAC-AES-128: CMAC-AES-128 key type.	NTP symmetric key types.

ts-ntp-type

Name	Type	Value	Description
ts-ntp-type	union	inet:host string	Specifying NTP IP address(es)/ LOCL.

ts-operation-type

Name	Type	Value	Description
ts-operation-type	enumeration	create: Set clock server and server key index. modify: Change a clock server's key index.	Indicates the action that must be taken for tsclockserver.

ts-timezone-type

Name	Type	Value	Description
ts-timezone-type	enumeration	Africa/Abidjan Africa/Accra Africa/Addis_Ababa Africa/Algiers Africa/Asmara Africa/Bamako Africa/Bangui Africa/Banjul Africa/Bissau Africa/Blantyre Africa/Brazzaville Africa/Bujumbura Africa/Cairo Africa/Casablanca Africa/Ceuta Africa/Conakry Africa/Dakar Africa/Dar_es_Salaam Africa/Djibouti Africa/Douala Africa/El_Aaiun Africa/Freetown Africa/Gaborone Africa/Harare Africa/Johannesburg Africa/Kampala Africa/Khartoum Africa/Kigali Africa/Kinshasa Africa/Lagos Africa/Libreville Africa/Lome Africa/Luanda Africa/Lubumbashi Africa/Lusaka Africa/Malabo Africa/Maputo Africa/Maseru Africa/Mbabane Africa/Mogadishu Africa/Monrovia Africa/Nairobi Africa/Ndjamena Africa/Niamey Africa/Nouakchott Africa/Ouagadougou Africa/Porto-Novo Africa/Sao_Tome Africa/Tripoli Africa/Tunis Africa/Windhoek	Timezone region and city.
		America/Adak America/Anchorage America/Anguilla America/Antigua America/Araguaina America/Argentina/Buenos_Aires	

tunnel-ha-type

Name	Type	Value	Description
tunnel-ha-type	enumeration	main: The 'main' tunnel is the primary tunnel of the extension-tunnel configured between local source-ip-address and destination remote-ip-address of the circuit[s]. The main tunnel is used to carry traffic over the WAN in standard operating conditions. The traffic may failover to either 'local-backup' or 'remote-backup' during the window when HCL is in progress. The failover traffic on either of 'local-backup' and 'remote-backup' and subsequent failback is determined by the fact as to which switch is under HCL i.e local or remote. local-backup: The 'local-backup' tunnel is the backup for the main tunnel for local HCL and uses the local-ha-ip-address and remote-ip-address of the circuit as the endpoint. Note the local-ha-ip-address and local-ip-address must be configured on the different DP's. This is to ensure that the traffic can sequentially failover to other DP during HCL sequence. The 'local-backup' tunnel will be seen as 'remote-backup' tunnel on the remote switch because of symmetric configuration. remote-backup: The 'remote-backup' tunnel is the backup for the main tunnel for remote HCL and uses the source-ip-address and remote-ha-ip-address of the circuit as the endpoint. Note the remote-ip-address and remote-ha-ip-address must be configured on the different DP's on the remote switch.This is to ensure that the traffic can sequentially failover to other DP of remote switch during HCL. The 'remote-backup' tunnel will be seen as 'local-backup' tunnel on the remote switch because of symmetric configuration.	The different HA support for extension-tunnel for HCL

tunnel-load-level

Name	Type	Value	Description
tunnel-load-level	enumeration	default: Specifies the 'default' method for load balancing method for the extension tunnel. failover: Specifies the 'failover' method for load balancing method for the extension tunnel. spillover: Specifies the 'spillover' method for load balancing method for the extension tunnel.	Load balancing method for the extension tunnel.

uri

Name	Type	Value	Description
uri	string		The uri type represents a Uniform Resource Identifier (URI) as defined by STD 66. Objects using the uri type MUST be in US-ASCII encoding, and MUST be normalized as described by RFC 3986 Sections 6.2.1, 6.2.2.1, and 6.2.2.2. All unnecessary percent-encoding is removed, and all case-insensitive characters are set to lowercase except for hexadecimal digits, which are normalized to uppercase as described in Section 6.2.2.1. The purpose of this normalization is to help provide unique URIs. Note that this normalization is not sufficient to provide uniqueness. Two URIs that are textually distinct after this normalization may still be equivalent. Objects using the uri type may restrict the schemes that they permit. For example, 'data:' and 'urn:' schemes might not be appropriate. A zero-length URI is not a valid URI. This can be used to express 'URI absent' where required. In the value set and its semantics, this type is equivalent to the Uri SMIv2 textual convention defined in RFC 5017.

usb-filename-type

Name	Type	Value	Description
usb-filename-type	string	length: 0..128 pattern: [a-zA-Z0-9_.\s-]*	The name of file on the USB device.

usb-path-type

Name	Type	Value	Description
usb-path-type	string	length: 1..128 pattern: [-~]+	The path to the file.

user-config-access-hours-type

Name	Type	Value	Description
user-config-access-hours-type	uint16	range: 1..24	Hours in 24h format

user-config-access-minutes-type

Name	Type	Value	Description
user-config-access-minutes-type	uint16	range: 1..60	Minutes

user-config-login-user-name-type

Name	Type	Value	Description
user-config-login-user-name-type	string	length: 1..32 pattern: ([a-zA-Z_][0-9a-zA-Z._]*)	Specifies the name of the remote user account.

user-config-role-type

Name	Type	Value	Description
user-config-role-type	string	length: 0|4..16 pattern: ^$|[a-z]{4,16}	Specifies the account's role. Displays an empty string if there is no role configured.

user-config-user-name-type

Name	Type	Value	Description
user-config-user-name-type	string	length: 1..32 pattern: ([a-zA-Z][0-9a-zA-Z._]{0,31})	Specifies the name of the account.

user-password-type

Name	Type	Value	Description
user-password-type	string	length: 1..40 pattern: [-~]{1,40} pattern: : modifier: invert-match	Specifies a password for the account.

user-port-number-type

Name	Type	Value	Description
user-port-number-type	int32	range: -1..3400	Fibre Channel port user port number.

uuid

Name	Type	Value	Description
uuid	string	pattern: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}	A Universally Unique IDentifier in the string representation defined in RFC 4122. The canonical representation uses lowercase characters. The following is an example of a UUID in string representation: f81d4fae-7dec-11d0-a765-00a0c91e6bf6

validation-mode-type

Name	Type	Value	Description
validation-mode-type	enumeration	basic: Basic mode strict: Strict mode	X509 validation mode selection

ve-interface-type

Name	Type	Value	Description
ve-interface-type	string	pattern: ((([1][0-2]|[1-9])/(1[6-9]|2[0-9]|3[0-5]))|((0)/(2[4-9]|3[0-9]|4[0-3]))|((0)/(1[2-5])))	VE interface of an Extension blade or system. The slot and port number of the port (slot/port).

vendor-name-type

Name	Type	Value	Description
vendor-name-type	string	length: 1..255 pattern: [-~]{1,255}	The vendor name for the media.

vendor-revision-type

Name	Type	Value	Description
vendor-revision-type	string	length: 0..255 pattern: [-~]{0,255}	The vendor revision number for the media.

virtual-channel-type

Name	Type	Value	Description
virtual-channel-type	int32	range: 0..128	Virtual Channel number

virtual-fabric-action-type

Name	Type	Value	Description
virtual-fabric-action-type	enumeration	addIf: Add virtual fabric list in existing user configuration. deleteIf: Delete virtual fabric list from existing user configuration.	Add/Delete virtual fabric for user configuration

virtual-fabric-configuration-group

Description

This common model is used for both user-config and ldap-role-map module

Parameters

Table 243: virtual-fabric-configuration-group

Name	Type	Description
home-virtual-fabric	home-virtual-fabric-type	Specifies the account's home Virtual Fabric. If the value is not configured in ldap-role-map, GET will fetch '0' value.
chassis-access-role	user-config-role-type	Specifies the account's access permissions regarding chassis-level commands.

virtual-fabric-role-id-type

Name	Type	Value	Description
virtual-fabric-role-id-type	string		Specifies the Virtual Fabrics to be added to the LDAP role Format is <role>=<virtual-fabric-id-list> where role is the LDAP role and virtual-fabric-id-list is a list of comma-seperated virtual fabric(s). Example strings include: o user=5 o admin=9-14 o newuserA=1,2,3 o newuserB=1,11-21

voltage-type

Name	Type	Value	Description
voltage-type	decimal64	range: 0..9999.99	The transceiver voltage of the media.

wwn-type

Name	Type	Value	Description
wwn-type	string	length: 0..24 pattern: ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){7})?	A 64-bit identifier, with a 60-bit value preceded by a 4-bit Network_Address_Authority Identifier, used to identify entities in Fibre Channel (e.g., Nx_Port, node, F_Port, or Fabric). Fibre channel WWN (16 hex digits with the colon separator). For example: 11:22:33:44:55:66:77:88

xpath1.0

Name	Type	Value	Description
xpath1.0	string		This type represents an XPATH 1.0 expression. When a schema node is defined that uses this type, the description of the schema node MUST specify the XPath context in which the XPath expression is evaluated.

yang-identifier

Name	Type	Value	Description
yang-identifier	string	length: 1..max pattern: [a-zA-Z_][a-zA-Z0-9\-_.]* pattern: .|..|[^xX].*|.[^mM].*|..[^lL].*	A YANG identifier string as defined by the 'identifier' rule in Section 12 of RFC 6020. An identifier must start with an alphabetic character or an underscore followed by an arbitrary sequence of alphabetic or numeric characters, underscores, hyphens, or dots. A YANG identifier MUST NOT start with any possible combination of the lowercase or uppercase character sequence 'xml'.

zero-based-counter32

Name	Type	Value	Description
zero-based-counter32	yang:counter32		The zero-based-counter32 type represents a counter32 that has the defined 'initial' value zero. A schema node of this type will be set to zero (0) on creation and will thereafter increase monotonically until it reaches a maximum value of 2^32-1 (4294967295 decimal), when it wraps around and starts increasing again from zero. Provided that an application discovers a new schema node of this type within the minimum time to wrap, it can use the 'initial' value as a delta. It is important for a management station to be aware of this minimum time and the actual time between polls, and to discard data if the actual time is too long or there is no defined minimum time. In the value set and its semantics, this type is equivalent to the ZeroBasedCounter32 textual convention of the SMIv2.

zero-based-counter64

Name	Type	Value	Description
zero-based-counter64	yang:counter64		The zero-based-counter64 type represents a counter64 that has the defined 'initial' value zero. A schema node of this type will be set to zero (0) on creation and will thereafter increase monotonically until it reaches a maximum value of 2^64-1 (18446744073709551615 decimal), when it wraps around and starts increasing again from zero. Provided that an application discovers a new schema node of this type within the minimum time to wrap, it can use the 'initial' value as a delta. It is important for a management station to be aware of this minimum time and the actual time between polls, and to discard data if the actual time is too long or there is no defined minimum time. In the value set and its semantics, this type is equivalent to the ZeroBasedCounter64 textual convention of the SMIv2.

zone-member-type

Name	Type	Value	Description
zone-member-type	union	zoning-name-type fibrechannel:wwn-type domain-index-type	A union of the three different zone member types (WWN, domain,index, or zone alias).

zone-operation-action-type

Name	Type	Value	Description
zone-operation-action-type	enumeration	expunge: Use this action type to expunge a zone object. In addition to deleting the object, the expunge operation will also remove the object from the member lists of all other objects. Expunge operations can be performed on cfgs, zones, aliases, WWNs, or 'D,I' members. This action stages the modifications in the defined-configuration. For the staged modifications to become persistent, the user must save the changes using one of the following brocade-zone module leafs that reside under the effective-configuration container: 'cfg-action-v2': Specify 'save' to save the pending changes. 'cfg-name': Specify the name of configuration to become enabled. It is recommended that the save be performed within the configured fabric-lock-timeout duration specified in the brocade-fibrechannel-configuration module under the zone-configuration container. rename: Use this action type to rename a zone object. In addition to renaming the object, this action will also replace all occurrences of the renamed zone object in the entire defined zone database. Rename operations can be performed on cfgs, zones, aliases. This action stages the modifications in the defined-configuration. For the staged modifications to become persistent, the user must save the changes using one of the following brocade-zone module leafs that reside under the effective-configuration container: 'cfg-action-v2': Specify 'save' to save the pending changes. 'cfg-name': Specify the name of configuration to become enabled. It is recommended that the save be performed within the configured fabric-lock-timeout duration specified in the brocade-fibrechannel-configuration module under the zone-configuration container.	The zone operation action type field. This field must be supplied for any RPC zone operation.

zone-type-string-type

Name	Type	Value	Description
zone-type-string-type	enumeration	zone: A zone that is not a peer zone. user-created-peer-zone: A peer zone that was created by a non-target device. target-created-peer-zone: A peer zone that was created by a target device.	The zone type. This field must be supplied for any peer zone operation.

zone-type-type

Name	Type	Value	Description
zone-type-type	uint8	range: 0|1|2	The zone type. This field must be supplied for any peer zone operation. The values are: zone : 0 user-created peer zone : 1 target-created peer zone : 2 Deprecated: Use 'zone-type-string-type' instead.

zoning-name-type

Name	Type	Value	Description
zoning-name-type	string	length: 1..64 pattern: ([0-9a-zA-Z][0-9a-zA-Z_^$-]{0,63})	Fibre Channel zoning names can consist of alphanumeric, '$', '^', '-', and '_' characters with a maximum length of 64 characters and are case sensitive. The pattern checking verifies that the specified name consists of alphanumeric characters and underline characters only. Names must begin with an alphanumeric character.

References

This section provides additional information about the FOS REST API and the resource and YANG data models.

REST API Description

REpresentational State Transfer (REST) is a style of software architecture for networked applications.

REST is based on the following principles:

- Everything is a resource. These resources are exposed through a universal resource indicator (URI).
- Every resource is identified by a unique identifier.
- Communication is through simple and uniform interfaces by representation.
- Clients can cache the responses. The responses must define themselves as cacheable or not cacheable to prevent the client from sending inappropriate data in response to further requests.
- All interactions are stateless; this means that all the information necessary to service the request must be contained in the URL, query parameters, body, or headers.

REST itself is not a standard, but it prescribes the use of standards such as HTTP, URL, and XML/HTML. RESTCONF is a REST-like protocol running over HTTP, used to access data defined in YANG (Yet Another Next Generation) model language using datastores defined in the Network Configuration Protocol, or NETCONF (RFC 6241).

Resources

A resource is an object with a type, associated data, relationships to other resources, and a set of methods that operate on it.

A resource has a set of methods such as GET, HEAD, OPTIONS, POST, PATCH, and DELETE are defined for it. Resources can be grouped into collections (in the YANG model, a collection is represented as a “List” statement). Each collection is homogeneous (a collection contains only one type of resource) and unordered. Refer to the table below for a more complete description of the methods.

Table 244: Supported FOS REST API Methods

Method	Description
DELETE	Deletes the specified resource.
GET	Retrieves the representation of the resource (for example, “base, configuration”) including the metadata.
HEAD	Retrieves the metadata of the resource identified in the request. The response to this operation contains only the headers and an empty response body.
OPTIONS	Retrieves the allowed methods for the resource identified in the request. The response to this operation contains the headers and an empty response body. The operations allowed on the resource are returned in the response “Allow” header.
PATCH	Specifies an ordered list of edits to be applied to the target datastore by the RESTCONF server.
POST	Creates a new resource in the resource location identified by the URI specified in the request. The URI of the new resource is returned in the response “Location” header.

Base Resources

Base resource elements represent high-level resources in the system and are categorized by media type.

In the FOS REST API implementation the only defined media type is XML. The XML representation is defined in *The YANG 1.1 Data Modeling Language* (RFC 7950), and is supported by the “application/yang-data+xml” media type.

The entry point container in the resource model is “/rest”, and all fields and sub-resources with the same resource type are defined in the namespace “http://<device_ID>/rest”, where <device_ID> is the IP address for the device. Refer to Logging In and Out for more information on the device ID.

The FOS REST API does not support actions on base resources or on the first-level containers of modules.

Resources and the YANG Model

Three types of resources are supported to represent configuration data and YANG-RPC operations: base resource, configuration resource, and YANG-RPC operations resource.

Resources can exist inside or outside a collection. Resources that are outside a collection are known as singleton resources (in the YANG model, these are represented in “Container” statements). Collections are resources themselves. For example, resources defined in the YANG model are physical interface, switches, zones, and so on.

The following figure illustrates the YANG resource model.

Figure 5: YANG Resource Models

Relationship between the YANG and Resource Data Models

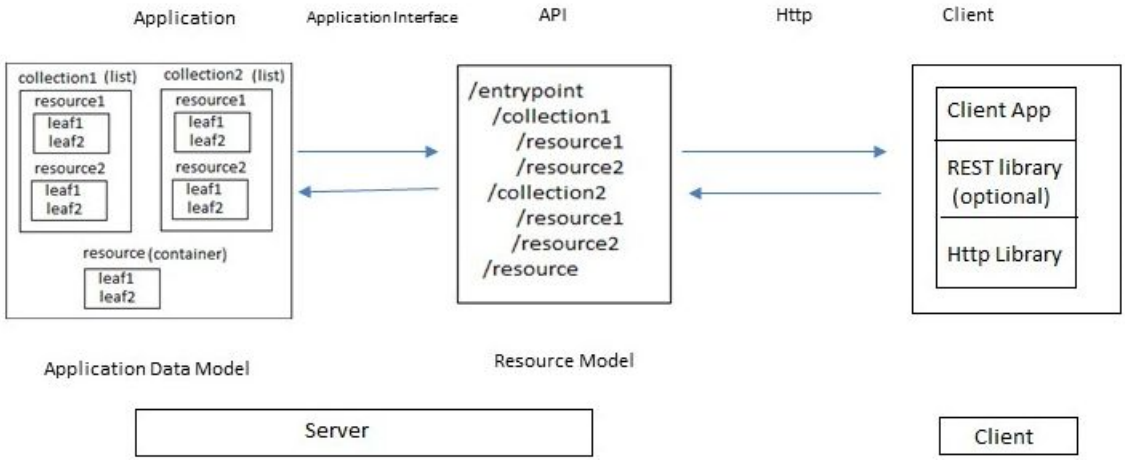
The resource data model is based on the YANG data model. All top-level containment statements such as “List” and “Container” that are present in the YANG data model are the resources, with a few exceptions.

Table 245: Top-Level Containment Statements

Statement	Description
List	A collection of resources that contains the same type of resources, which are ordered. Entries such as a Container statement inside a List statement are also a resource.
Container	A singleton resource or a group of resources of different types.
Leaf	Leaf statements inside a List or Container resource are the attributes of these sources. A Leaf is a subresource of a List or Container. That is, a Leaf cannot be identified without referencing its parent List or Container resource.

The following diagram illustrates the relationship between YANG and Resource data.

Figure 6: YANG and Resource Data Model Relationship



Uniform Resource Identifiers

A uniform resource identifier (URI) is a link used to identify a specific resource. It is the only means for clients and servers to exchange resource representations.

URIs have two parts:

- Base URI—The base URI is specific to the Fabric OS server. All URIs accessing the same server use the same base URI.
- Request URI—The request URI is the URI used to perform a GET, POST, PATCH, DELETE, HEAD, or OPTIONS operation.

In this example of a URI, the text in bold is the base URI, and the remaining portion is the request URI:

`http://10.10.10.122:80/rest``/running/brocade-fabric/fabric-switch`

Common practice for URI illustration is to compact the base URI as <base_URI>, as shown in the following example:

`<base_URI>``/running/brocade-extension-tunnel/`

NOTE
URIs are case-sensitive.

URI Structure

The hierarchical structure of the URI is used to support those resources that are defined in the YANG model by the List and Container statement elements.

The URI path conveys a resource model that is similar to the YANG model, with each forward-slash-separated path segment corresponding to a unique resource within the model's hierarchy (using the following syntax: <base_URI>/path1/path2/{key1},{key2}/path3/...). For example, the URI `/rest/running/brocade-interface/fibrechannel` identifies the Fibre Channel interfaces as target resources. In this example, anything in the URI from the path element ("…/brocade-interface") onward represents YANG model resources.

- rest – The entry point.
- running – Represents the running configuration datastore.
- brocade-interface – Represents all interfaces present in the running configuration.
- fibrechannel – Represents all Fibre Channel interfaces present in the running configuration.

There are some additional restrictions that apply specifically to the FOS REST API. See FOS REST API URI Restrictions for details.

URI Case

FOS REST API enables you to use kebab-case or camelCase when entering a URI path and request body. Kebab-case uses hyphens to maximize white space between multiple words, while the camelCase does not use any white space. Instead, camelCase uses capitalization to distinguish between multiple words in a REST API request. CamelCase starts each new word with an uppercase letter.

REST API 9.1.x and earlier supports kebab-case only for the REST API. Kebab-case uses hyphens to separate words (such as `/rest/tunning/brocade-maps/monitoring-system-matrix`).

REST API 9.2.x and later enables you to use kebab-case or camelCase for the REST API. CamelCase uses capitalization to distinguish between multiple words (such as `rest/running/brocadeMaps/monitoringSystemMatrix`).

To use camelCase, you must include the following key and value in the headers:

- Key – Camel-Case-Mode
- Value – on

When camelCase is enabled, the URI path and request body must use camelCase. If you use kebab-case and camelCase is on, you receive a 'bad request' response.

To turn off camelCase and only use kebab-case, make sure that Camel-Case-Mode is not included in the headers or include the following key and value in the headers:

- Key – Camel-Case-Mode
- Value – off

URI Encoding

The following rules apply to encoding URI content directly in the request. Encoding is not supported for content in request bodies.

- A key that contains a forward slash (/) must be encoded as “%2f”. For example, the port value “0/2” is encoded as `0%2f2` .
- The delimiter between adjacent keywords in URIs is a comma (,). This is encoded as `%2C` .
- There are other scenarios that could require encoding, for example IPsec policy names support the special characters such as @, %, and *. For example, an IPsec policy named “ABC%123” would be encoded as: `/rest/running/brocade-extension-ipsec-policy/extension-ipsec-policy/policy-name/ABC%25123` , with the percent symbol encoded as “%25”.

For additional information on HTML URI encoding values, refer to https://en.wikipedia.org/wiki/Percent-encoding.

Base URI

The base URI (https://host:port/rest/) is the entry point to access and manage all resources that are defined in the system. The port is the default HTTPS port (443). It is used to identify the base resource and to retrieve the first-level child resources that belong to the base resource.

> **NOTE**
> A leaf attribute can also be present in a URI to identify the exact resource. For example, the URI `<base URI>/rest/running/brocade-interface/fibrechannel/name/<name>/operational-status` identifies the operational-status resource of the specified fibrechannel interface.

Top-Level URIs

The URI identifies the first-level resource in its hierarchy with the media type in its request, for example, `<base URI>/config/running`.

> **NOTE**
> The FOS REST API does not support any methods on top-level URIs. For example, you cannot make any request to `<base URI>/rest/running/brocade-zone`.

XML Resource Representation

In the FOS REST API, a resource is represented in XML as an XML element. Sub-resources are encoded as sub-elements to the resource element.

In the XML representation of a list resource, the keys are always present and encoded first, and leafs are properties of the resource. Single-valued resource properties are encoded as sub-elements to the resource element, with the value encoded as character data in the sub-element. Refer to the module descriptions elsewhere in this publication for illustrations.

The following example shows the XML representation of the “operational-status” resource for port 0/0. Notice that it uses “%2f” for the slash character. **Request URI**

```
GET https://10.10.10.10/rest/running/brocade-interface/fibrechannel/name/0%2f0/operational-status
```

Request Body No request body is required. **Response Body** When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel>   <== container resource
        <name>0/0</name> <== leaf "name"
        <operational-status>3</operational-status> <== leaf "operational-status"
    </fibrechannel>
</Response>
```

The “name” and “operational-status” leafs are sub-elements of the “fibrechannel” resource.

JSON Resource Representation

The Fabric OS REST API supports JSON (JavaScript Object Notation) resource representation. For JSON, a resource is represented as a single top-level object or array. Sub-resources are encoded as sub-elements (arrays objects) to the resource element.

In the JSON representation of a list resource, the keys are always present and encoded first, and leafs are properties of the resource. Single-valued resource properties are encoded as sub-elements to the resource element, with the value encoded as character data in the sub-element. Refer to the module descriptions elsewhere in this publication for illustrations.

Object: An object can be defined as any structured data with name-value pairs. An object begins and ends with curly braces (`{ }`). The name and associated value are separated by a colon (for example, `"name": "10:00:c4:f5:7c:16:8e:9a"`) and name-value pairs are separated by a comma. In the example below, a switch object displays:

```
{
     "Response": {
     "fibrechannel-switch": {
     "name": "10:00:c4:f5:7c:16:8e:9a",
     "domain-name": "englab.brocade.com.",
```

```
        …
        …
        "principal": 1,
        "ip-address": {
        "ip-address": "10.10.10.10"
        },
        "model": "162.0",
        "firmware-version": "v8.2.0_01",
        "vf-id": 128,
        "fabric-user-friendly-name": "xyz",
        "ag-mode": 1
        }
      }
}
```

Array: An array is a container object that holds a collection of values. An array begins and ends with square brackets (`[ ]`). Array elements can be a string, number, object, array, boolean or null.

> **NOTE**
> In Fabric OS 8.2.1b, for consistency in the FOS REST API implementation all list URIs are returned using the array format, which begins and ends with square brackets ([ ]), even if the list only has a single member.

In the example below, “entry-name” displays an array of string elements within “zone” object.

```
{
     "zone": {
     "zone-name": "z",
     "zone-type": 0,
     "member-entry": {
     "entry-name": [
          "10:00:8c:7c:ff:5e:a3:00",
          "20:05:00:11:0d:5b:01:00",
          "2c:82:8c:7c:ff:5e:a3:00"
         ]
          }
     }
   }
}
```

String: A string is zero or more Unicode character within quotes. In the example below, the string is contain within quotes:

```
{   "domain-name": "englab.brocade.com."  }
```

Number: A number in JSON must be integer or floating point value. In the example below, "ag-mode" displays as 1.

```
{    "ag-mode": 1 }
```

Boolean: A boolean values is either true or false without quotes. In the example below, "is-enabled" displays as true.

```
{    "is-enabled": true  }
```

Null: A null value can be represented by null or empty quotes (`""`). In the examples below, "sample-name" displays as null.

```
{    "sample-name": null }


{       "sample-name": "" }
```

The following example shows the JSON representation of the "name", "domain-id", "fcid", "user-friendly-name", "enabled-state", "up-time", "domain-name", "principal", "ip-address", "model", "firmware-version", "vf-id", "fabric-user-friendly-name", and "ag-mode" as sub-elements of the "fibrechannel-switch" resource (top-level object). **Request URI**

```
GET https://10.10.10.10/rest/running/brocade-switch/fibrechannel-switch
```

Request Body No request body is required. **Response Body** When the operation is successful, the response has a message body similar to the following, and a “200 OK” status in the headers.

```
{
    "Response": {
        "fibrechannel-switch": {
            "name": "10:00:c1:f2:3c:10:4e:5a",
            "domain-id": "1",
            "fcid": "16776193",
            "user-friendly-name": "switch1",
            "enabled-state": 2,
            "up-time": 4846,
            "domain-name": "eng.brcd.com",
            "principal": 0,
            "ip-address": {
                "ip-address": "10.10.10.10"
            },
            "model": "162.0",
            "firmware-version": "v8.2.0_01",
            "vf-id": 128,
            "fabric-user-friendly-name": "xyz",
            "ag-mode": 1
        }
    }
}
```

The "name", "domain-id", "fcid", "user-friendly-name", "enabled-state", "up-time", "domain-name", "principal", "ip-address", "model", "firmware-version", "vf-id", "fabric-user-friendly-name", and "ag-mode" leafs are sub-elements of the “fibrechannel-switch” resource.

Error Reporting

A Brocade-specific ResourceID (XML) is encoded in to the standard RESTCONF “errors” structure, so that it is returned as part as any failure report. The error type determines the error-path value. Object- and operation-level errors use the root URI command in the error path, while attribute-level errors return the object name and attribute as part of the error path value.

If there is an XML parsing error in the request, the request will return without processing any of the objects in the request. If objects are processed and any errors found, the request will be returned with the object identifier along with keys in the error-path. You can then identify the exact error path and correct any errors before applying the right payload and resubmitting the request.

The following is the YANG tree diagram for error data:

```
+---- errors
     +---- error*
```

```
        +---- error-type       enumeration
        +---- error-tag        string
        +---- error-app-tag?   string
        +---- error-path?      instance-identifier
        +---- error-message?   string
        +---- error-info?
```

Refer to the sample error logs for details on how different single or multiple errors are returned.

Error Returned Due to an Invalid POST Request

This example shows the error returned when an invalid POST request is made.

URI

```
POST http://10.10.10.10/rest/running/brocade-fibrechannel-configuration/zone-
configuration/node-name-zoning-enabled
```

Request body

No request body is required.

Response body

Notice the error-message and error-info fields. They been rendered in bold for the example; they will not be bold in the actual response.

```
<?xml version="1.0" ?>
<errors xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
    <error>
        <error-type>application</error-type>
        <error-tag>operation-not-supported</error-tag>
        <error-app-tag>Error</error-app-tag>
        <error-path>/rest/running/brocade-fibrechannel-configuration/zone-configuration/node-name-
zoning-enabled</error-path>
        <error-message>Method not supported</error-message>
        <error-info>
            <error-code>25</error-code>
            <error-module>rest</error-module>
        </error-info>
    </error>
</errors>
```

Existing IP Address Error

This example shows the error returned when a multiple IP address configuration is tried using POST and the IP addresses are already configured.

URI

```
http://10.10.10.10/running/brocade-interface/extension-ip-interface
```

Request Body

```
<extension-ip-interface>
   <name>4/0</name>
   <dp-id>0</dp-id>
   <ip-address>10.10.0.11</ip-address>
   <ip-prefix-length>24</ip-prefix-length>
</extension-ip-interface>
<extension-ip-interface>
   <name>4/0</name>
   <dp-id>1</dp-id>
   <ip-address>10.10.0.10</ip-address>
   <ip-prefix-length>24</ip-prefix-length>
</extension-ip-interface>
```

Response Body

Notice the value for the error-message fields. This has been rendered in bold for the example; it will not be bold in the actual response.

```
<?xml version="1.0"?>
<errors xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
  <error>
    <error-type>protocol</error-type>
    <error-tag>Operation-failed</error-tag>
    <error-app-tag>Error</error-app-tag>
    <error-path>/extension-ip-interface/name/4/0/</error-path>
    <error-message>IP Address already configured.
IP Address 10.10.0.11 already configured on port 4/ge0. IP interface creation failed.</error-
message>
    <error-info>
      <error-code>54</error-code>
      <error-module>extension-service</error-module>
    </error-info>
  </error>
  <error>
    <error-type>protocol</error-type>
    <error-tag>Operation-failed</error-tag>
    <error-app-tag>Error</error-app-tag>
    <error-path>/extension-ip-interface/name/4/0/</error-path>
    <error-message>IP Address already configured.
IP Address 10.10.0.10 already configured on port 4/ge0. IP interface creation failed.</error-
message>
    <error-info>
      <error-code>54</error-code>
      <error-module>extension-service</error-module>
    </error-info>
  </error>
```

```
</errors>
```

Unsupported Platform Error

This example shows the error returned when a GET request using brocade-interface/gigabitethernet is executed on a platform that does not support a Gigabit Ethernet interface.

URI

```
http://10.10.10.10/rest/running/brocade-interface/gigabitethernet
```

Request body

No request body is required.

Response body

Notice the value for the error-message field. This has been rendered in bold for the example; it will not be bold in the actual response.

```
<?xml version="1.0" ?>
<errors xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
    <error>
        <error-type>application</error-type>
        <error-tag>unknown-element</error-tag>
        <error-app-tag>Error</error-app-tag>
        <error-path>/rest/running/brocade-interface/gigabitethernet</error-path>
        <error-message>Invalid resource identifier</error-message>
        <error-info>
            <error-code>-1</error-code>
            <error-module>cal</error-module>
        </error-info>
    </error>
</errors>
```

Login Errors

Login errors (other than a flawed ID/password combination) are usually due to either trying to log in using the HTTPS protocol when a security certificate is not installed on the switch (or using HTTP when a certificate is installed) or having an invalid session key.

This example shows the error returned when a request is made that is invalid because the session key is not valid. This is usually because the session has not been correctly logged in to the switch or the session has timed out.

URI

```
http://10.10.10.10/rest/login
```

Request body

No request body is required.

Response body

Notice the error-message and error-info fields. They are rendered in bold for the example; they will not be bold in the actual response.

```
<?xml version="1.0" ?>
<errors xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
    <error>
```

```
            <error-type>application</error-type>
            <error-tag>operation-failed</error-tag>
            <error-app-tag>Error</error-app-tag>
            <error-path>/rest/login</error-path>
            <error-message>Invalid user in the session key</error-message>
            <error-info>
                <error-code>17</error-code>
                <error-module>auth</error-module>
            </error-info>
        </error>
    </errors>
```

Deprecated and Obsolete Resources in Fabric OS 9.2.0

The following sections detail the deprecated and obsolete resources for the initial release of Fabric OS 9.2.0.

Deprecated

The following section details the resources that are deprecated in Fabric OS 9.2.0. The deprecated resources are listed under the URI where they are located.

- *<base_URI>*/rest/running/brocade-access-gateway/port-group
  - load-balancing-mode-enabled (Use load-balancing-mode-enabled-v2 instead.)
  - multiple-fabric-name-monitoring-mode-enabled (Use multiple-fabric-name-monitoring-mode-enabled-v2 instead.)
  - failover-enabled (Use failover-enabled-v2 instead.)
  - failback-enabled (Use failback-enabled-v2 instead.)
- *<base_URI>*/rest/running/brocade-access-gateway/f-port-list
  - f-port-info/ro login-exceeded (Use login-exceeded-v2 instead.)
- *<base_URI>*/rest/running/brocade-access-gateway/policy
  - port-group-policy-enabled (Use port-group-policy-enabled-v2 instead.)
  - auto-policy-enabled (Use auto-policy-enabled-v2 instead.)
- *<base_URI>*/rest/running/brocade-extension-ipsec-policy/extension-ipsec-policy
  - restart-ike-sessions (Use brocade-operation-restart instead.)
- *<base_URI>*/rest/running/brocade-extension-tunnel/extension-tunnel
  - admin-enabled (Use admin-enabled-v2 instead.)
  - fast-write-enabled (Use fast-write-enabled-v2 instead.)
  - tape-read (Use tape-read-v2 instead.)
  - tape-write (Use tape-write-v2 instead.)
  - tunnel-status (Use tunnel-status-v2 instead.)
  - ipsec-enabled (Use ipsec-enabled-v2 instead.)
  - ficon (Use ficon-v2 instead.)
  - ficon-xrc-acceleration (Use ficon-xrc-acceleration-v2 instead.)
  - ficon-tape-write-acceleration (Use ficon-tape-write-acceleration-v2 instead.)
  - ficon-tape-read-acceleration (Use ficon-tape-read-acceleration-v2 instead.)
  - ficon-tin-tir-emulation (Use ficon-tin-tir-emulation-v2 instead.)
  - ficon-device-acknowledgement-emulation (Use ficon-device-acknowledgement-emulation-v2 instead.)
  - ficon-read-block-id-emulation (Use ficon-read-block-id-emulation-v2 instead.)
  - ficon-teradata-read-acceleration (Use ficon-teradata-read-acceleration-v2 instead.)
  - ficon-teradata-write-acceleration (Use ficon-teradata-write-acceleration-v2 instead.)
  - ip-extension (Use ip-extension-v2 instead.)
  - compression-tunnel (Use compression-tunnel-v2 instead.)
  - compression-protocol/fc-compression (Use fc-compression-v2 instead.)
  - compression-protocol/ip-compression (Use ip-compression-v2 instead.)
- *<base_URI>*/rest/running/brocade-extension-tunnel/extension-tunnel-statistics
  - flow-status (Use flow-status-v2 instead.)
- *<base_URI>*/rest/running/brocade-extension-tunnel/extension-ciruit

  - admin-status (Use admin-status-v2 instead.)
  - dscp/priority-control (Use priority-control-v2 instead.)
  - dscp/fc-priority-high (Use fc-priority-high-v2 instead.)
  - dscp/fc-priority-medium (Use fc-priority-medium-v2 instead.)
  - dscp/fc-priority-low (Use fc-priority-low-v2 instead.)
  - dscp/ip-priority-high (Use ip-priority-high-v2 instead.)
  - dscp/ip-priority-medium (Use ip-priority-medium-v2 instead.)
  - dscp/ip-priority-low (Use ip-priority-low-v2 instead.)
  - circuit-status (Use circuit-status-v2 instead.)
  - arl-algorithm-mode (Use arl-algorithm-mode-v2 instead.)
- *<base_URI>*/rest/running/brocade-extension-tunnel/extension-ciruit-statistics
  - flow-status (Use flow-status-v2 instead.)
- *<base_URI>*/rest/running/brocade-fdmi/hba
  - boot-bios-enabled (Use boot-bios-enabled-v2 instead.)
- *<base_URI>*/rest/running/brocade-fibrechannel-configuration/switch-configuration
  - area-mode (Use area-mode-v2 instead.)
- *<base_URI>*/rest/running/brocade-fibrechannel-configuration/f-port-login-settings
  - enforce-login (Use enforce-login-string instead.)
- *<base_URI>*/rest/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch
  - base-switch-enabled (Use base-switch-enabled-v2 instead.)
  - logical-isl-enabled (Use logical-isl-enabled-v2 instead.)
  - ficon-mode-enabled (Use ficon-mode-enabled-v2 instead.)
- *<base_URI>*/rest/running/brocade-fibrechannel-switch/fibrechannel-switch
  - up-time (Use brocade-chassis/chassis-uptime instead.)
- *<base_URI>*/rest/running/brocade-interface/fibrechannel

  - auto-negotiate (Use ro auto-negotiate-v2 instead.)
  - e-port-disable (Use e-port-disable-v2 instead.)
  - n-port-enabled (Use n-port-enabled-v2 instead.)
  - d-port-enable (Use d-port-enable-v2 instead.)
  - persistent-disable (Use persistent-disable-v2 instead.)
  - qos-enabled (Use qos-enabled-v2 instead.)
  - compression-configured (Use compression-configured-v2 instead.)
  - encryption-enabled (Use encryption-enabled-v2 instead.)
  - compression-active (Use compression-active-v2 instead.)
  - encryption-active (Use encryption-active-v2 instead.)
  - target-driven-zoning-enable (Use target-driven-zoning-enable-v2 instead.)
  - sim-port-enabled (Use sim-port-enabled-v2 instead.)
  - mirror-port-enabled (Use mirror-port-enabled-v2 instead.)
  - credit-recovery-enabled (Use credit-recovery-enabled-v2 instead.)
  - credit-recovery-active (Use credit-recovery-active-v2 instead.)
  - fec-active (Use fec-active-v2 instead.)
  - csctl-mode-enabled (Use csctl-mode-enabled-v2 instead.)
  - fault-delay-enabled (Use fault-delay-enabled-v2 instead.)
  - vc-link-init (Use vc-link-init-v2 instead.)
  - isl-ready-mode-enabled (Use isl-ready-mode-enabled-v2 instead.)
  - rscn-suppression-enabled (Use rscn-suppression-enabled-v2 instead.)
  - npiv-enabled (Use npiv-enabled-v2 instead.)
  - npiv-flogi-logout-enabled (Use npiv-flogi-logout-enabled-v2 instead.)
  - ex-port-enabled (Use ex-port-enabled-v2 instead.)
  - fec-enabled (Use fec-enabled-v2 instead.)
  - via-tts-fec-enabled (Use via-tts-fec-enabled-v2 instead.)
  - port-autodisable-enabled (Use port-autodisable-enabled-v2 instead.)
  - trunk-port-enabled (Use trunk-port-enabled-v2 instead.)
- *<base_URI>*/rest/running/brocade-interface/fibrechannel-statistics
  - reset-statistics (Use brocade-operation-port instead.)
- *<base_URI>*/rest/running/brocade-interface/gigabitethernet
  - enabled-state (Use enabled-state-v2 instead.)
  - persistent-disable (Use persistent-disable-v2 instead.)
- *<base_URI>*/rest/running/brocade-interface/gigabitethernet-statistics
  - reset-statistics (Use brocade-operation-port instead.)
- *<base_URI>*/rest/running/brocade-maps/maps-config
  - quiet-time
- *<base_URI>*/rest/running/brocade-maps/rule
  - quiet-time
- *<base_URI>*/rest/running/brocade-maps/monitoring-system-matrix
  - is-quiet-time-supported
  - minimum-quiet-time
- *<base_URI>*/rest/running/brocade-media/media-rdp
  - media-speed-capability (Use media-speed-capability-v2 instead.)
  - remote-media-speed-capability (Use remote-media-speed-capability-v2 instead.)
- *<base_URI>*/rest/running/brocade-zone/effective-configuration

  - cfg-action (Use cfg-action-v2 instead.)
  - default-zone-access (Use default-zone-access-v2 instead.)

Obsolete

The following section details the resources made obsolete in Fabric OS 9.2.0. The obsolete resources are listed under the URI where they are located.

- *<base_URI>*/rest/running/brocade-extension-tunnel
  - wan-statistics (Use wan-statistics-v1 instead.)
- *<base_URI>*/rest/running/brocade-fibrechannel-configuration/chassis-config-settings
  - ezserver-enabled
- *<base_URI>*/rest/running/brocade-fibrechannel-diagnostics/fibrechannel-diagnostics
  - electrical-loopback-test
  - optical-loopback-test
- *<base_URI>*/rest/running/brocade-firmware
  - firmware-config
- *<base_URI>*/rest/running/brocade-logging/raslog
  - message-text
- *<base_URI>*/rest/running/brocade-maps/rule
  - un-quarantine-clear
- *<base_URI>*/rest/running/brocade-security/security-certificate-generate
  - unit-name

Revision History

FOS-92x-REST-API-RM100; 28 April 2023

Initial document version.

Documentation Legal Notice

This notice provides copyright and trademark information as well as legal disclaimers.

Copyright © 2021–2022. Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

The product described by this document may contain open source software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products or to view the licensing terms applicable to the open source software, please download the open source attribution disclosure document in the Broadcom Support Portal. If you do not have a support account or are unable to log in, please contact your support provider for this information.

BROADCOM®