Brocade Fabric OS REST API Reference Manual, 8.2.x

Reference Manual
December 27, 2021

Table of Contents

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.

About This Document

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 Fabric OS 8.2.x.

Although many different software and hardware configurations are tested and supported by Broadcom Inc. for Fabric OS 8.2.x, documenting all possible configurations and scenarios is beyond the scope of this document.

Fabric OS support for the Brocade Analytics Monitoring Platform (AMP) device depends on the specific version of the software running on that platform. For more information, refer to the Brocade Analytics Monitoring Platform documentation and release notes.

Brocade Gen 5 (16G) Fixed-Port Switches

- Brocade 6505 Switch
- Brocade 6510 Switch
- Brocade 6520 Switch
- Brocade M6505 blade server SAN I/O module
- Brocade 6542 blade server SAN I/O module
- Brocade 6543 blade server SAN I/O module
- Brocade 6545 blade server SAN I/O module
- Brocade 6546 blade server SAN I/O module
- Brocade 6547 blade server SAN I/O module
- Brocade 6548 blade server SAN I/O module
- Brocade 6558 blade server SAN I/O module
- Brocade 7840 Extension Switch

Brocade Gen 5 (16G) Directors

For ease of reference, Brocade chassis-based storage systems are standardizing on the term “director.” The legacy term “backbone” can be used interchangeably with the term “director.”

- Brocade DCX 8510-4 Director
- Brocade DCX 8510-8 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

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.
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.
request URI	A request URI is the URI used to perform a REST HTTP request such as GET, POST, PUT, DELETE, HEAD or OPTIONS.

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

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

Online	Telephone
For nonurgent issues, the preferred method is to log on to myBroadcom at https://www.broadcom.com/mybroadcom. (You must initially register to gain access to the Customer Support Portal.) Once there, select **Customer Support Portal** > **Support Portal**. You will now be able to navigate to the following sites: • **Knowledge Search**: Clicking the top-right magnifying glass brings up a search bar. • **Case Management**: The legacy MyBrocade case management tool (MyCases) has been replaced with the Fibre Channel Networking case management tool. • **DocSafe**: You can download software and documentation. • **Other Resources**: Licensing Portal (top), SAN Health (top and bottom), Communities (top), Education (top).	Required for Severity 1 (critical) issues: Please call Fibre Channel Networking Global Support at one of the numbers listed at https://www.broadcom.com/support/fibre-channel-networking/.

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

- OEM/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/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/solution provider.

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.

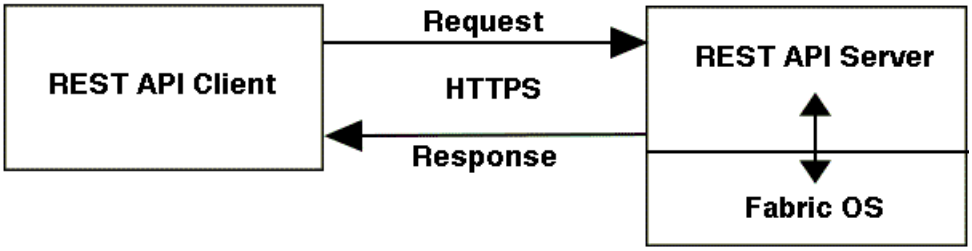
This API is referred to as the FOS REST API and is enabled by default in Fabric OS 8.2.0 and later. It follows the RESTCONF protocol defined in IETF RFC 8040 and defines data using the YANG 1.1 (Yet Another Next Generation) data modeling language as defined in IETF RFC 7950.

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, which is the largest zone database size supported on Director-only fabrics.
- 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 that applies only to REST transactions. 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. Pending FOS API REST zoning operations time out after 5 minutes. After 5 minutes 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 5-minute timeout restriction applies only to REST zoning transactions, and is not user-configurable.

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 the following 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.

Command Throttling

Fabric OS sets limits on how often a FOS REST API session can invoke Fabric OS subsystems via command throttling; this prevents a REST session from consuming excessive memory or CPU resources.

The three attributes used to configure throttling are:

- Idle time
- Sampling time
- Number of REST requests allowed within the sampling time

See REST Session Configuration for the default attribute values and information on configuring these attributes.

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.

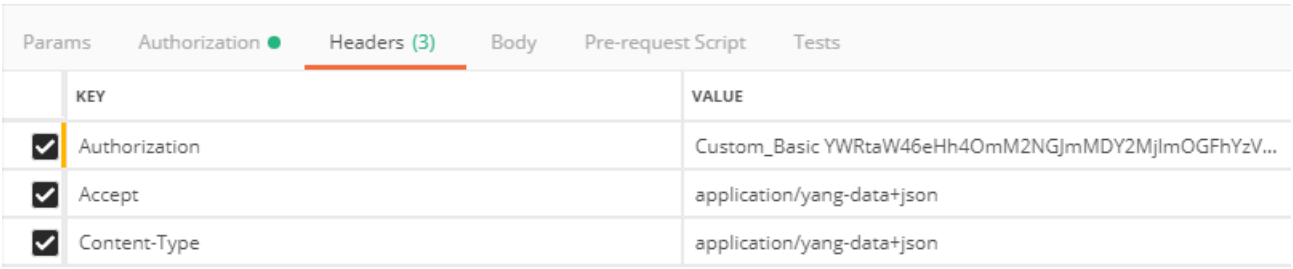
```
POST https://10.10.10.10/rest/login
POST https://[10:10:10:eb:1a:b7:77:bc]:443/rest/login
```

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

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.

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 2: 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 3: 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
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.

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.

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

FOS REST API session-less operation requests 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 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=1.30.0.

For example, the resource API version for Fabric OS 9.1.0 is 1.50.0 (major.minor.patch). The resource API version for Fabric OS 9.0.1 is 1.40.0. The resource API version for Fabric OS 9.0.0a is 1.40.0. The resource API version for Fabric OS 8.2.1b is 1.30.0.

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 9.1.x.

Table 4: Module Level Version

Module Name	Module Version
Auth-token	1.0.0
brocade-access-gateway	1.30.0
brocade-application-server	1.0.0
brocade-chassis	1.20.0
brocade-extension	2.0.0
brocade-extension-ip-route	1.10.10
brocade-extension-ipsec-policy	1.0.10
brocade-extension-tunnel	1.50.0
brocade-fabric	2.20.0
brocade-fabric-traffic-controller	1.10.0
brocade-fdmi	1.20.0
brocade-fibrechannel-configuration	1.20.0
brocade-fibrechannel-diagnostics	2.20.0
brocade-fibrechannel-logical-switch	2.10.0
brocade-fibrechannel-routing	1.10.0
brocade-fibrechannel-switch	3.20.0
brocade-fibrechannel-trunk	1.10.20
brocade-ficon	1.0.0

brocade-firmware	1.0.0
brocade-fru	2.0.0
brocade-interface	3.20.0
brocade-license	1.20.20
brocade-lldp	1.0.10
brocade-logging	1.30.0
brocade-login-response	1.0.0
brocade-management-ip-interface	1.0.0
brocade-maps	2.0.0
brocade-media	2.0.0
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-extension	1.0.0
brocade-operation-fabric	1.0.0
brocade-operation-factory-reset	1.0.0
brocade-operation-firmwaredownload	1.10.0
brocade-operation-license	1.10.20
brocade-operation-lldp	1.10.0
brocade-operation-management-ethernet-interface	1.0.0
brocade-operation-pcie-health	1.0.10
brocade-operation-port	1.0.0
brocade-operation-portchannel	1.0.0
brocade-operation-port-decommission	1.0.0
brocade-operation-reboot	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	1.0.0
brocade-operation-security-fabric-wide-policy-distribute	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-show-status	1.10.0

brocade-operation-supportlink	1.10.10
brocade-operation-supportsave	1.10.10
brocade-operation-time	1.0.0
brocade-operation-traffic-optimizer	1.0.0
brocade-operation-usb-delete	1.0.0
brocade-operation-zone	1.10.0
brocade-security	3.0.0
brocade-snmp	1.20.0
brocade-supportlink	1.10.0
brocade-time	1.20.0
brocade-traffic-optimizer	1.0.0
brocade-usb	1.0.0
brocade-zone	2.10.20
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>
```

Deprecated and Obsolete Resources

The following sections detail the deprecated and obsolete resources for the FOS REST API.

Deprecated Resources

- Fabric OS 8.2.0a
  - fcid (brocade-fabric, brocade-interface, brocade-fibrechannel-switch) – Use the *fcid-hex* leaf.
- Fabric OS 8.2.1b
  - brocade-extension-tunnel/admin-enabled – Use the *is-admin-enabled* leaf to enable or disable and *admin-status* to obtain the current status.
  - enabled-state
    - brocade-fibrechannel-switch – Use the *is-enabled-state* leaf to configure and the *operational-status* leaf to obtain the current state of the switch.
    - brocade-interface/fibrechannel – Use the *is-enabled-state* leaf.
- Fabric OS 9.0.0
  - brocade-chassis

    - part-number – Use the brocade-fru/list wwn/*part-number* leaf.
    - vendor-serial-number – Use the brocade-fru/list wwn/*vendor-serial-number* leaf.
    - vendor-part-number – Use the brocade-fru/list wwn/*vendor-part-number* leaf.
    - vendor-revision-number – Use the brocade-fru/list wwn/*vendor-revision-number* leaf.
  - brocade-fabric/path-count – Use the *brocade-fibrechannel-switch/topology-domain path-count* leaf.
  - brocade-fibrechannel-switch
    - errors-detected-local – Use the *failure-report-local-details* leaf-list.
    - errors-detected-remote – Use the *failure-report-remote-details* leaf-list.
  - brocade-interface
    - port-type – Use the `port-type-string` leaf
    - pod-license-status – Use the `pod-license-state` leaf.
  - brocade-maps/quiet-time-clear – Use the `quiet-time` leaf.
  - brocade-snmp
    - access-control – Use the *brocade-security/ipfilter-policy* list.
    - security-get-level – Use the *security-get-level-string* leaf.
    - security-set-level – Use the *security-set-level-string* leaf.
  - brocade-zone/zone-type – Use the *zone-type-string* leaf.

- Fabric OS 9.0.1
  - brocade-maps/port-loss – Use the *loss-of-sync* leaf.

- Fabric OS 9.1.0
  - brocade-access-gateway/online-status – Use the *operational-status* leaf.
  - brocade-access-gateway/reliable-status – Use the *reliable-status-string* leaf.
  - brocade-extension/rtt – Use the *tcp-rtt* leaf.
  - brocade-fdmi/supported-speed – Use the *supported-protocol-speeds* leaf list.
  - brocade-fdmi/current-port-speed – Use the *current-protocol-speed* leaf.
  - brocade-fibrechannel-logical-switch/default-switch-status – Use the *default-switch* leaf.
  - brocade-fibrechannel-switch/ip-address – Use the *static-ip-addresses* and *stateful-ip-addresses* leafs in brocade-management-ip-interface.
  - brocade-fibrechannel-switch/ip-static-gateway-list – Use the *static-ip-gateways*, *dhcp-gateway*, and *stateless-ip-gateways* leafs in brocade-management-ip-interface.
  - brocade-fibrechannel-switch/subnet-mask – Use the *static-subnet-mask* and *dhcp-subnet-mask* leafs in brocade-management-ip-interface.
  - brocade-fibrechannel-switch/ag-mode – Use the *ag-mode-string* leaf).
  - brocade-fru/ip-address-list – Use the *static-ip-addresses* and *stateful-ip-addresses* containers.
  - brocade-fru/ip-gateway-list – Use the*static-ip-gateways*, *dhcp-gateway*, and *stateless-ip-gateways* leafs in brocade-management-ip-interface.
  - brocade-fru/subnet-mask – Use the*static-subnet-mask* and *dhcp-subnet-mask* leafs.
  - brocade-brocade-interface/fibrechannel/operational-status – Use the *operational-status-string* leaf.
  - brocade-brocade-interface/fibrechannel/speed – Use the *protocol-speed* leaf.
  - brocade-brocade-interface/fibrechannel/maxspeed – Use the *max-protocol-speed* leaf.
  - brocade-brocade-interface/fibrechannel/octet-speed-combo – Use the *octet-speed-combo-string* leaf.
  - brocade-brocade-interface/fibrechannel/long-distance – Use the *long-distance-string* leaf.
  - brocade-brocade-interface/fibrechannel/los-tov-mode-enabled – Use the *los-tov-mode-enabled-string* leaf.
  - brocade-interface/gigabitethernet/speed – Use the *protocol-speed* leaf.
  - brocade-interface/portchannel/speed – Use the*protocol-speed* leaf.
  - brocade-name-server/link-speed – Use the *protocol-speed* leaf.

Obsolete Resources

- Fabric OS 9.0.0
  - brocade-interface/fibrechannel
    - g-port-locked
    - non-dfe-enabled
    - rate-limit-enabled
  - brocade-name-server/share-area
  - brocade-security
    - /sshutil-key/passphrase
    - /ipfilter-rule/traffic-type
- Fabric OS 9.1.0
  - brocade-fru/fru-time-detail-group
  - brocade-media
    - remote-media-voltage-alert
    - remote-media-temperature-alert
    - remote-media-tx-bias-alert
    - remote-media-tx-power-alert
    - remote-media-rx-power-alert
    - high-alarm
    - low-alarm
    - high-warning
    - low-warning
    - alert-type

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>v8.2.1_bld40</firmware-version>
        <vf-id>10</vf-id>   <== Virtual fabric ID
        <ip-address>10.38.66.91</ip-address>
        <fcip-address>0.0.0.0</fcip-address>
        <ipv6-address>2620:100:4:fa01:227:f8ff:fe0f:b9f0</ipv6-address>
        <principal>0</principal>
    </fibrechannel-switch>
</Response>
```

Fabric OS REST Session Configuration

The `mgmtapp` Fabric OS CLI command 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, and configuring keepalive for the REST session. 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 configuration values (interface state, effective protocol, and HTTP state), 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 Configuration:
    Interface State: Enabled
    Effective Protocol: HTTP only
    HTTP State: Enabled
    Session Count: 3
  HTTPS Configuration:
    KeepAlive : Disabled
    KeepAliveTimeout : 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`.

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.

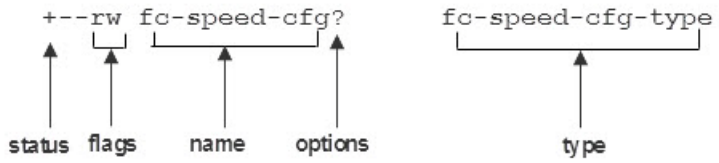
YANG Module Overview

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 5: 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.

Additional FOS REST API Data Types

The following sections describe additional data types supported by the FOS REST API.

Additional General Data Types

The following table describes additional data types supported by the FOS REST API. Additional details about these data types can be found in the associated .yang file (for example, brocade-zone.yang). This file is located in the same folder as the module .yang files.

Table 6: General Data Types

Name	Type	Range/Pattern/Length	Description
air-flow-direction-type (brocade-fru.yang)	string	forward (non-portside exhaust)| non-portside exhaust|reverse (non-portside intake)|non-portside intake|not available	The air flow direction of the fan.
alert-type (brocade-media.yang)	int16	-32767–32767	The high or low alert type for the media.
authentication-data (brocade-extension-ipsec-policy.yang)	union	“preshared-key” or “pki-key-pair”	The authentication data format. The value depends on the profile used, either “preshared-key” or “pki-key-pair”.
brocade-hex-string-type	string	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).
capable-speed-type (brocade-media.yang)	uint16	1–1000	The supported transmission speeds of the media.

Name	Type	Range/Pattern/Length	Description
comments-type (brocade-fibrechannel-diagnostics.yang)	string	|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	Additional textual information elaborating on the completion status when a test is skipped or fails. It returns an empty string if no comments are available.
congestion-detection-signal-cycles-type (brocade-fabric-traffic-controller.yang)	uint16	0|1–999	The signal threshold - congestion detection cycle count: • 0 = Not supported. • 1 to 999 = The number of cycles at the indicated signaling scale used to detect congestion signals.
congestion-detection-signal-type (brocade-fabric-traffic-controller.yang)	enumeration	"none", "warning", or "warning-and-alarm"	The congestion detection signal capability.
congestion-detection-signal-scale-type (brocade-fabric-traffic-controller.yang)	enumeration	"none", "seconds", "milliseconds", "microseconds", or "nanoseconds"	The signal threshold - congestion detection scale in seconds, milliseconds, microseconds, or nanoseconds.
congestion-state-type (brocade-fabric-traffic-controller.yang)	enumeration	"none", "credit-stalled", "lost-credit", or "oversubscribed"	The congestion status.
current-type (brocade-media.yang)	decimal64	0–999.99	The transceiver current of the media.
date-code-type (brocade-media.yang)	string	[-~]{1,32}	The vendor's manufacturing date code.
domain-index-type (brocade-zone.yang)	string	([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". The value for domain is between 1 and 239 and the value for index is between -1 and 65535.
domain-port-type	string	([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]))	Domain/port zone members appear as “domain,port”. The value for domain is from 1 through 239 and the value for port is from 0 through 65535.
els-descriptor-type (brocade-fabric-traffic-controller.yang)	enumeration	"none", "fpin-peer-congestion", "fpin-link-integrity", "fpin-congestion", or "fpin-delivery"	The ELS + ELS-descriptor pairs that are exchanged via the Register Diagnostic Functions (RDF) ELS.

Name	Type	Range/Pattern/Length	Description
error-stats-type (brocade-fibrechannel-diagnostics.yang)	enumeration	"none", "generic", "enc-in", "crc-err", "trunc-frame", "frame-too-long", "bad-eof", "enc-out", "bad-os", "c3-discard", "rx-c3-timeout", "tx-c3-timeout", "unroutable", "unreachable", "other-discard", "link-failure", "sync-loss", "signal-loss", "prim-seq-proto", "inv-tx-word", "invalid-crc", "delimiter-err", "addr-id-err", "link-reset-in", "link-reset-out", "port-ols-in", "port-ols-out", "inv-arb", "single-credit-loss", "multi-credit-loss", "fec-uncorrected", "pcs-block-error ", or "unknown-error	The error statistics on the port.
extension-ipsec-profile-name (brocade-extension-ipsec-policy.yang)	enumeration	“preshared” or “pki”	An Extension IPsec policy name. This is either “preshared” or “pki”.
fc-speed-type	uint64	0|1000000000|2000000000| 4000000000|8000000000| 100000000000|160000000000| 320000000000|400000000000| 53000000000|64000000000	The nominal port bandwidth. • 0 = Auto-negotiated speed • 1000000000 = 1Gb/s • 2000000000 = 2Gb/s • 4000000000 = 4Gb/s • 8000000000 = 8Gb/s • 10000000000 = 10Gb/s • 16000000000 = 16Gb/s • 32000000000 = 32Gb/s • 40000000000 = 40Gb/s • 53000000000 = 53Gb/s • 64000000000 = 64Gb/s
fcr-device-state (brocade-fibrechannel-routing.yang)	enumeration	"configured", "initializing", "exist", or "imported"	The state of the device.
ge-interface-type	string	(([1][0-2]|[0-9])/(1[0-7]|[0-9]))	GE interface of an Extension blade or system in the form of the slot and port number of the port (slot/port).
interface-type (brocade-media.yang)	string	6–16 ((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]))	The network interface name in 3-tuple canonical format: interface-id/slot/port. The interface IDs include fc, ge, and te.
lag-port-timeout (brocade-interface.yang)	enumeration	"short" or "long"	The timeout (short or long) configured for the port channel member.
lldp-mandatory-tlv-name-type (brocade-lldp.yang)	enumeration	"chassis-id", "port-id", or "time-to-live"	The LLDP mandatory TLV names.
lldp-optional-tlv-name-type (brocade-lldp.yang)	enumeration	"dcbx", "fcoe-app", "fcoe-lls", "dot1", "dot3", "mgmt-addr", "port-desc", "sys-cap", "sys-desc", or "sys-name"	The LLDP optional TLV names.

Name	Type	Range/Pattern/Length	Description
logical-path-operational-states (brocade-ficon.yang)	enumeration	"Oper" or Reset"	The operational state for a logical path.
media-power-type (brocade-media.yang)	decimal64	0–9999.99	The power type of the media.
operational-type	uint16	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

Name	Type	Range/Pattern/Length	Description
pki-key-pair (brocade-extension-ipsec-policy)	string	1–32 characters, matching the following pattern: Excludes “/”. The modifier is “invert-match”.	This is a certificate name. For a “pki” profile, this specifies the local key pair name to use for IKE authentication. This operand is required for logging in using a PKI profile.
portchannel-type (brocade-interface.yang)	enumeration	"static" or "dynamic"	Whether the port channel is static or dynamic.
port-type-string-type (brocade-interface.yang)	enumeration	"e-port", "g-port", "universal-port", " f-port", "l-port", "fcoe-port", "ex-port", "d-port", " sim-port", "af-port", "ae-port", "ve-port", "ethernet-port", "flex-port", "n-port", "mirror-port", "encryption-support-port", "loopback-port", or "unknown-port"	The supported Fibre Channel port types.
preshared-key (brocade-extension-ipsec-policy)	string	16–64 characters, matching the following pattern: '[a-zA-Z0-9~@%_+:\[\]][a-zA-Z0-9~@%_+:\[\]\-]{15,63}';	For a “preshared” profile, this specifies the preshared-key to be used for authentication. This operand is required with shared-key authentication.
result-type (brocade-fibrechannel-diagnostics.yang)	string	NOT STARTED|PASSED|SKIPPED|FAILED|IN PROGRESS|RESPONDER|STOPPED|UNKNOWN	The completion status of a diagnostic test.
speed-type	uint64	1000000000|10000000000|40000000000	The Gigabit Ethernet port bandwidth. • 1000000000 = 1Gb/s • 10000000000 = 10Gb/s • 40000000000 = 40Gb/s
temperature-type (brocade-media.yang)	int16	-128–1000	The temperature of the media.
tunnel-load-level	enumeration	default|failover|spillover	The load balancing method for the extension tunnel.
ve-interface-type	string	((([1][0-2]|[1-9])/(1[6-9]|2[0-9]|3[0-5]))|((0)/(2[4-9]|3[0-9]|4[0-3])))	VE interface of an Extension blade or system in the form of the slot and port number of the port (slot/port).
vendor-name-type (brocade-media.yang)	string	0–255	The vendor name for the media.
vendor-revision-type (brocade-media.yang)	string	0–255	The vendor revision number for the media.
voltage-type (brocade-media.yang)	decimal64	0–999.99	The transceiver voltage of the media.
zone-member-type (brocade-zone.yang)	union	This data type is composed of the following data types: • zoning-name-type • fibrechannel:wwn-type • domain-index-type	The zone member composed of the following three zone member types: • WWN • Domain-Index • zone alias
zone-operation-action-type (brocade-operation-zone.yang)	enumeration	"expunge"	The zone operation action type field. You must complete this field for any RPC zone operation.
zone-type-string-type (brocade-zone.yang)	enumeration	"zone", "user-created-peer-zone", or "target-created-peer-zone	The zone type. This field must be supplied for any peer zone operation.

Name	Type	Range/Pattern/Length	Description
zone-type-type (brocade-zone.yang)	uint8	0|1|2	The zone type. • 0 = Default (any zone that is not a peer zone) • 1 = User-created peer zone • 2 = Target-created peer zone
zoning-name-type (brocade-zone.yang)	string	1–64 characters, matching the following pattern: ([0-9a-zA-Z_\-^$]{1,64})	A counter64 that has zero as its defined initial value.

Fibre Channel Data Types

The following table describes additional Fibre Channel data types supported by the FOS REST API. For additional details about Fibre Channel data types, refer to the fibrechannel-yang-types.yang file.

Table 7: Fibre Channel Data Types

Name	Type	Range/Pattern/Length	Description
brocade-hex-string-type	string	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).
deskew-type	uint16	1–255	The time in 10ns units. This is used by trunking for fibrechannel ports.
domain-id-type	uint32	0–239	A Fibre Channel switch domain identifier. Note that a value of zero indicates a segmented link.
fabric-id-type	uint32	1–128	A fabric ID (FID).
fcid-type **Deprecated**: Use 'fcid-hex-string-type' instead.	uint32	[0-9A-F]{12}	A Fibre Channel ID (FCID) in decimal format. This is a 3-byte (24-bit) field used to route frames through a FC network.
fcid-hex-string-type	string	0[xX][0-9a-fA-F]	A Fibre Channel ID (FCID) for an Nx_Port (FCID), hexadecimal format. The format allows for six hexadecimal digits prefixed by '0x' or '0X'. For example: 0x010200

Name	Type	Range/Pattern/Length	Description
fc4-type-type	string	(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)	The FC4 type. • Unknown/Reserved <hexadecimal value> = (0xnn) • 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)
fc4-features-type	string	(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.
fibrechannel_extended_isl	N/A	N/A	Indicates that Extended ISL (XISL) use is supported on this platform. XISL use requires VF Mode to be enabled.
fibrechannel_extension_platform	N/A	N/A	Indicates that the Extension platform is supported.
fibrechannel_virtual_fabric_platform	N/A	N/A	Indicates that Virtual Fabric mode is enabled.
octet-member-index-type	int8	0–7	Identifies the member in the octet for a trunk.
percentage-type	string	([0-9][0-9].[0-9][0-9]|100.00)	A percentage string up to two decimal places.
portchannel_platform	N/A	N/A	Indicates that Port Channels are supported on this platform.
port-type-string-type	string	0[xX][0-9a-fA-F]{2}|n-port|nl-port|f/nl-port|nx-port|f-port|fl-port|e-port|b-port|a-port	A Fibre Channel port type identifier. • Unknown/Reserved <hexadecimal value> = (0xnn) • n-port = (0x01) • nl-port = (0x02) • f/nl-port = (0x03) • nx-port = (0x7F) • f-port = (0x81) • fl-port = (0x82) • e-port = (0x84) • b-port = (0x85) • a-port = (0x86)

Name	Type	Range/Pattern/Length	Description
slot-port-name-type	string	(([1-9]|1[0-9])/([1][0-9][0-9][0-9]|[1-9][1-9][0-9]|[1-9][1-9]|[1-9]))	The slot and port number (slot/port) of the interface.
speed-type	string	(|bit-[0-9]{1,2}|speed-1-gfc|speed-2-gfc|speed-10-gfc|speed-4-gfc|speed-8-gfc|speed-16-gfc|speed-32-gfc|speed-20-gfc|speed-40-gfc|speed-128-gfc|speed-64-gfc|speed-256-gfc|speed-not-established|speed-10-ge|speed-40-ge|speed-100-ge|speed-25-ge|speed-50-ge|speed-400-ge)*	Fibre Channel transmission speeds. • bit-n = (Unknown/Reserved <bit position>) • speed-1-gfc = (Mask Value (hex): 0000 0001) • speed-2-gfc = (Mask Value (hex): 0000 0002) • speed-10-gfc = (Mask Value (hex): 0000 0004) • speed-4-gfc = (Mask Value (hex): 0000 0008) • speed-8-gfc = (Mask Value (hex): 0000 0010) • speed-16-gfc = (Mask Value (hex): 0000 0020) • speed-32-gfc = (Mask Value (hex): 0000 0040) • speed-20-gfc = (Mask Value (hex): 0000 0080) • speed-40-gfc = (Mask Value (hex): 0000 0100) • speed-128-gfc = (Mask Value (hex): 0000 0200) • speed-64-gfc = (Mask Value (hex): 0000 0400) • speed-256-gfc = (Mask Value (hex): 0000 0800) • speed-not-established = (Mask Value (hex): 0000 8000) • speed-10-ge = (Mask Value (hex): 0001 0000) • speed-40-ge = (Mask Value (hex): 0002 0000) • speed-100-ge = (Mask Value (hex): 0004 0000) • speed-25-ge = (Mask Value (hex): 0008 0000) • speed-50-ge = (Mask Value (hex): 0010 0000) • speed-400-ge = (Mask Value (hex): 0020 0000)
time-generated-type	uint32	1–2147483646	The time since the start of an epoch in seconds. An epoch is an instant in time chosen as the origin of a particular measurement block.
user-port-number-type	int32	-1–3400	A Fibre Channel port user port number.
wwn-type	string	([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){7})	A Fibre Channel WWN. This is 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). Typically shown as 16 uppercase hexadecimal characters with a colon separating each tuple.

FRU Module Data Types

The following table describes additional FRU data types supported by the FOS REST API. Additional details about these data types can be found in the associated brocade-fru-types.yang file. This file is located in the same folder as the module .yang files.

Table 8: FRU Module Data Types

Name	Type	Range/Pattern/Length	Description
firmware-version-type	string	1–255 [-~]{1,255}	A human readable string identifying the firmware version running on the switch.
ip-gateway-type	union	[*]	The IP address of an IP router that can route packets to the destination IP address. Enter an '*' (asterisk) to retrieve route entries that are created internally by the system.
manufacturer-type	string	1–63 [-~]*	The name of the organization responsible for producing the blade.
part-number-type	string	1–14 [-~]{1,14}	The part number assigned by the organization responsible for producing or manufacturing the physical element.
serial-number-type	string	1–15 [-~]{1,15}	A printable ASCII string that specifies the serial number of the chassis.

ietf-inet Data Types

The following table describes additional ietf-inet data types supported by the FOS REST API. For additional details ietf-inet data type refer to the ietf-inet-types.yang file. This file is located in the same folder as the module .yang files.

Table 9: ietf-inet Data Types

Name	Type	Range/Pattern/Length	Description
inet:as-number	uint32	N/A	The 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. Refer to RFC 1930, RFC 4271, RFC 4001, and RFC 6793.
inet:domain-name	string	1–253 characters	A domain name service (DNS) domain name. Refer to RFC 1123.
inet:dscp	uint8	0–63	A differentiated services code point (DSCP) that can 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. Refer to RFC 3289, RFC 2474, and RFC 2780.
inet:host	union	N/A	The host type represents either an IP address or a DNS domain name.
inet:ip-address	union	N/A	An IP address that is IP version neutral, where the format of the address indicates the IP version. This type supports scoped addresses by allowing zone identifiers in the address format. Refer to RFC 4007.
inet:ip-address-no-zone	union	N/A	An IP address that is IP version neutral, where the format of the address indicates the IP version. This type supports scoped addresses by allowing zone identifiers in the address format. Refer to RFC 4007.
inet:ipv4-address	uint16	N/A	An IP address in the IPv4 format. Refer to RFC 791.
inet:ipv4-address-no-zone	inet:ipv4-address	[0-9\.]*	An IPv4 address without a zone index. This type, derived from inet:ipv4-address type, may be used in situations where the zone is known from the context and hence no zone index is needed.

Name	Type	Range/Pattern/Length	Description
inet:ipv6-address	uint16	N/A	An IP address the IPv6 format. Refer to RFC 2460.
inet:ipv6-address-no-zone	inet:ipv4-address	[0-9a-fA-F:\.]*	An IPv6 address without a zone index. This type, derived from inet:ipv6-address type, may be used in situations where the zone is known from the context and hence no zone index is needed. Refer to RFC 4291, RFC 4007, and RFC 5952.
inet:ipv6-flow-label	uint32	0–1048575	The flow identifier or flow label in an IPv6 packet header that can 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. Refer to RFC 3595 and RFC 2460.
inet:ip-prefix	union	N/A	An IP prefix that is IP version neutral, where the format of the address indicates the IP version.
inet:ipv4-prefix	string	(([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 length of the 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.
inet:ipv6-prefix	string	((:|[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 length of the 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. Refer to RFC 5952.
inet:ip-version	enumeration	N/A	The IP protocol version. • **0** = unknown • **1** = ipv4 • **2** = ipv6

Name	Type	Range/Pattern/Length	Description
inet:port-number	uint16	0..65535	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 IANA (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. Refer to RFC 768, RFC 793, RFC 4960, RFC 4340, and RFC 4001.
inet:uri	string	N/A	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. Refer to RFC 3986, RFC 3305, and RFC 5017.

ietf-yang Data Types

The following table describes additional ietf-yang data types supported by the FOS REST API. For additional details on ietf-yang data types, refer to the ietf-yang-types.yang file. This file is located in the same folder as the module .yang files.

Table 10: ietf-yang Data Types

Name	Type	Range/Pattern/Length	Description
ietf:counter32	uint32	0–4294967296	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.
ietf:counter64	uint64	0–18446744073709551615	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.
ietf:date-and-time	string	\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d +)?' + '(Z|[\+\-]\d{2}:\d{2})	A profile using the ISO 8601 standard for representing dates and times using the Gregorian calendar. Refer to RFC 6991.
ietf:dotted-quad	string	(([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, such as, four octets written as decimal numbers and separated with the '.' (full stop) character.
ietf:gauge32	uint32	0–4294967296	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 in 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. Refer to RFC 2578.

Name	Type	Range/Pattern/Length	Description
ietf:gauge64	uint64	0–18446744073709551615	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 in decimal), 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. Refer to RFC 2856.
ietf:hex-string	string	([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.
ietf:mac-address	string	Six groups of two hexadecimal digits, separated by hyphens, matching the following pattern: ([0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5})	An IEEE 802.xx media access control address (MAC) address. The canonical representation uses lowercase characters.
ietf:object-identifier	string	(([0-1](\.[1-3]?[0-9]))|(2\.(0|([1-9]\d*)))(\.(0|([1-9]\d*)))*	An administratively assigned name 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, do not use this type to represent the SMIv2 OBJECT IDENTIFIER type; use the object-identifier-128 type instead. Refer to ISO/IEC 9834-1:2008.

Name	Type	Range/Pattern/Length	Description
ietf:object-identifier-128	object-identifier	\d*(\.\d*){1,127}	An object identifier restricted to 128 sub-identifiers. In the value set and its semantics, this type is equivalent to the OBJECT IDENTIFIER type of the SMIv2. Refer to RFC 2578.
ietf:phys-address	string	([0-9a-fA-F]{2}(:[0-9a-fA-F]{2})*)?	A media- or physical-level address represented as a sequence of octets, each octet represented by two hexadecimal numbers 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. Refer to RFC 2579.
ietf:timestamp	yang:timeticks	N/A	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. Refer to RFC 2579.
ietf:timeticks	uint32	0–4294967296	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.
ietf:uuid	string	[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. Refer to RFC 4122.
ietf:xpath1.0	string	N/A	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. Refer to "XPATH: XML Path Language (XPath) Version 1.0.

Name	Type	Range/Pattern/Length	Description
ietf:yang-identifier	string	[a-zA-Z_][a-zA-Z0-9\-_.]* .|..|[^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'. Refer to RFC 6020.
ietf:zero-based-counter32	uint32	0–4294967296	A counter32 that has zero as its defined initial value.
ietf:zero-based-counter64	uint64	0–18446744073709551615	See zero-based-counter64 above.

Logging Module Data Types

The following table describes additional logging module data types supported by the FOS REST API. Additional details about these data types can be found in the associated brocade-logging-types.yang file. This file is located in the same folder as the module .yang files.

Table 11: Logging Module Data Types

Name	Type	Range/Pattern/Length	Description
ascii-text-type	string	[-~]*	The type of text.
class-type	string	zone|security|configuration|firmware|fabric|ls|cli|maps	The type of log to be collected (such as fabric related log).
message-id-type	string	7 to 12 alphanumeric characters. [A-Z]{2,7}-[0-9]{4,4}	The unique identifier for the raslog message.
module-id-type	string	2 to 7 alphanumeric characters. [A-Z]{2,7}	The FOS module ID type.
severity-level-type	string	info|warning|error|critical	The severity level types.
time-24hr-type	string	5 numeric characters. [0-2][0-9]:[0-5][0-9]	The time in 24 hour clock in the format HH:MM.

MAPS Module Data Types

The following table describes additional Monitoring and Alerting Policy Suite (MAPS) data types supported by the FOS REST API. Additional details about these data types can be found in the associated brocade-maps-types.yang file. This file is located in the same folder as the module .yang files.

Table 12: MAPS Module Data Types

Name	Type	Range/Pattern/Length	Description
maps-types:credit-stall-state-type	enumeration	N/A	The available states on the FPI credit-stall congestion conditions. • frame-loss = The frame loss state. • perf-impact = The performance impact state. • medium = The medium state. • low = The low state. • info = The information state. • no-congestion = The no congestion state. • monitoring-paused = Monitoring is paused for the member.
maps-types:data-type-type	enumeration	N/A	The data type support. Each monitoring system supports different data types for thresholds. • unsigned-int32 = The unsigned integer 32 bits data type. • int32 = The integer 32 bits data type. • float = The float data type. • unsigned-int64 = The unsigned integer 64 bits data type. • enum = The enumeration data type.
maps-types:fru-state-type	enumeration	N/A	The FRU states. • faulty = The faulty state. • in = The plug in state. • out = The plug out state. • off = The power off state. • on = The power on state.

Name	Type	Range/Pattern/Length	Description
maps-types:maps-dashboard-category-type	enumeration	N/A	The dashboard category. Each rule can belong to only one category. • 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 FRU Health category enables you to define rules for field replaceable units (FRUs). • 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 monitors 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 monitors your system's temperature, flash usage, memory usage, and CPU. • fabric-performance-impact = The Fabric Performance Impact category monitors the current condition of the latency detected on E_Ports and F_Ports in a fabric over different time windows and uses this to determine the performance impact to the fabric and network • traffic-performance = The Traffic Performance category monitors flows.

Name	Type	Range/Pattern/Length	Description
maps-types:maps-data-unit	enumeration	N/A	The unit of the monitoring system. • CRCs • ITWs • LFs • loss-of-signal • errors • LRs • timeouts • milli-ampere • days • loss-of-signals • loss-of-synchronization • violations • ports • extension-flows • it-flows • certificates • segmentations • changes • logins • IOs • IOPS • MBps • % • hours • centigrade • bytes • micro-seconds • milli-volts • micro-watts • aborts • reserves • not-applicable
maps-types:maps-event-severity-type	string	1 to 10 characters. [info|warning|error|critical|^$]	The user-configured severity (such as warning, error, critical, or info). Each event has its own severity and is user editable. If the value is empty, the system assigns a severity to the rule. For more information about the default severity, refer to the *Brocade Fabric OS MAPS User Guide*.

Name	Type	Range/Pattern/Length	Description
maps-types:maps-generic-action-type	enumeration	N/A	The MAPS actions. You can enable one or more actions globally at the switch level or per rule. • port-fence = This action immediately takes ports offline, which might cause loss of traffic. • snmp-trap = This action generates a message (called a “trap”) that notifies a management station when specific events occur on a switch. • raslog = This action adds an entry to the switch event log for an individual switch. • sddq = This action moves the traffic destined to a port affected by device-based latency to a low-priority virtual channel. This action does not disable the port, but it reduces the effect of its latency on other flows in the fabric. • un-quarantine = This action releases the previously quarantined ports. • decommission = This action takes a port offline without loss of traffic. • port-toggle = This action temporarily disables a port and then re-enables it, allowing the port to reset and recover from some device based issues. • e-mail = This action sends information about the event to one or more specified e-mail addresses. The e-mail alert specifies the threshold and describes the event, much like an error message. • fms = This action MAPS sends a notification event information to the FICON management service. • vtap-uninstall = This action uninstalls vTAP feature if the mirrored frame count exceeds 250K IOPS and encryption is enabled on the 16Gb/s-capable ASIC. If encryption is not enabled on the ASIC, vTAP is not uninstalled. • re-balance = This action brings the port group state back to balance state. This may take 3 or more seconds. • sw-marginal = This action places the switch into a marginal operating state. • sw-critical = = This action places the switch into a down operating state. • sfp-marginal = This action places the SFP into a marginal operating state. • fpin = This action sends Fabric Performance Impact Notifications (FPIN) to registered end devices.

Name	Type	Range/Pattern/Length	Description
maps-types:maps-group-feature-type	string	1 to 32 characters. [node\-name|port\-wwn|^$]	The existing feature name for the group. If the value is empty, no user feature is assigned to the group. You can create user-defined groups matching the port name or node WWN. • node-wwn = The node WWN feature. • port-name = The port name feature.
maps-types:maps-group-name-type	string	1 to 32 alphanumeric characters. [0-9a-zA-Z_]*	The group name. The group name must be unique; it is not case sensitive and can contain up to 32 characters.
maps-types:maps-group-type-type	enumeration	N/A	• power-supply = The power supply group type. The device may have multiple power supplies and may be integrated with a fan. • fan = The fan group type. The device may have multiple fans may be integrated with power supplies. • fc-port = The FC port group type. • sfp = The FC SFP group type. The device may have different SFPs based on speed or vendor. • blade = The blade group type. This group type manages all blades including core, CP, or switch blades. • circuit = The FCIP circuit group type. • circuit-qos = The circuit QOS traffic group type. Note that each circuit can carry multiple QOS traffic. • temperature-sensor = The temperature sensor group type. • flash = The flash memory group type. • switch = The switch group type. • chassis = The chassis group type. • cpu = The CPU group type. • wwn = The WWN card group type. • flow = The flow group type. • tunnel = The tunnel group type. • tunnel-qos = The tunnel qos group type. • backend-port = The back end group type. • ge-port = The giga bit 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. • vtap-port = The vTAP port group type. • asic = The ASIC group type.

Name	Type	Range/Pattern/Length	Description
maps-types:maps-logical-operator-type	enumeration	N/A	The relational operation to be used in evaluating the condition. • 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.
maps-types:maps-monitoring-system-type	string	2 to 72 alphanumeric characters. [0-9a-zA-Z_]*	The monitoring system name (CRC, BLADE_STATE, ITW, and so on). A monitoring system is a value (measure or statistic) that can be monitored.
maps-types:maps-policy-name-type	string	1 to 32 alphanumeric characters. [0-9a-zA-Z_]*	The MAPS policy name. The name for the policy must be unique; it is case-sensitive and can contain up to 32 characters.
maps-types:maps-port-monitoring-state	enumeration	N/A	MAPS monitors each port and based on its operating condition it sets the state. • offline = The offline state. • online = The online state. • faulty = The faulty state. • marginal = The marginal state. A port can be in marginal state if the port does not have enough credits to operate. • error = The error state. A port cam be set to an error state if the port is not in operation due to hardware malfunction or security. • un-monitored = The un-monitored state.
maps-types:maps-quiet-time-unit-type	enumeration	N/A	The unit of quiet time. • minute = A minute. • hour = An hour. • day = A day.
maps-types:maps-rule-type	boolean	true|false	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.
maps-types:maps-rule-name-type	string	1 to 72 alphanumeric characters. [0-9a-zA-Z_]*	The rule name.
maps-types:oversubscription-state-type	enumeration	N/A	The available states on the FPI oversubscription conditions. • oversubscription = The oversubscription state. • no-oversubscription = The no oversubscription state. • monitoring-paused = Monitoring is paused for the member.

Name	Type	Range/Pattern/Length	Description
maps-types:ssp-state-type	enumeration	N/A	The switch status policy report state. The state gives the summary health of the switch and individual components. • unknown = The unkown state. • down = The down state - some service is already impacted. • marginal = The marginal state - if a FRU is in a marginal state, fix it or replace it with a new one. • healthy = The healthy state - normal operating condition.
maps-types:maps-time-base-type	enumeration	N/A	The time interval between two samples to be compared. • none = The time base is not applicable. • second = The samples are compared every second. • minute = The samples are compared every minute. • five-minute = The samples are compared every five minutes. • hour = The samples are compared every hour. • day = The samples are compared every day. • week = The samples are compared every week.
maps-types:monitoring-type-type	enumeration	N/A	The monitoring type (event based or poll based) for the monitoring system. • event-based = Event based monitoring. • poll-based = Poll based monitoring.
maps-types:threshold-value-type	string	1 to 32 alphanumeric characters. [0-9a-zA-Z,_]*	The threshold to be used in the condition.

Operations Module Data Types

The following table describes additional operations data types supported by the FOS REST API. Additional details about these data types can be found in the associated brocade-operations-types.yang file. This file is located in the same folder as the module .yang files.

Table 13: Operations Module Data Types

Name	Type	Range/Pattern/Length	Description
operation:message-id-type	uint32	1–maximum	The Remote Procedure Call (RPC) message. When asynchronous operation is triggered, it returns the random message identifier to associate it with the operation being triggered. The application could use this identifier to obtain the status of the operation in subsequent requests.
operation:show-status-group	N/A	N/A	The group of leafs associated with the operation. • show-status = The supportSave operation status based on the specified message ID. • message-id = The message ID associated with the supportSave operation. • operation = The operation (such as supportsave). • status = The exact state of the supportSave operation. When supportSave is triggered, it is added to the queue. The supportSave operation then goes into in-progress until all supportsave files are transferred to the remote server. Once the supportSave operation is complete, the status moves to the done, then to delivered until it is purged from the database. If any supportSave operational level or application level error occurs, it shows an error or application-error status respectively. • application-name = The name of the application that triggered the supportSave operation. • percentage-complete = The percentage complete of the operation.

Security Module Data Types

The following table describes additional security module data types supported by the FOS REST API. Additional details about these data types can be found in the associated brocade-security-types.yang file. This file is located in the same folder as the module .yang files.

Table 14: Security Module Data Types

Name	Type	Range/Pattern/Length	Description
aaa-authspec-type	string	1–128 radius;local|radius;localbackup| tacacs;local|tacacs;localbackup|ldap;local| ldap;localbackup|radius|tacacs+|ldap|local	The authentication mode for RADIUS, TACACS+, and LDAP.
aaa-encryption-algorithm-type	enumeration	none|aes256	The encryption algorithm type (none or aes256).

Name	Type	Range/Pattern/Length	Description
aaa-protocols-type	enumeration	chap|pap	The authentication protocols that are commonly used by both RADIUS and TACACS+.
aaa-timeout-type	uint16	1–30 seconds. The default response timeout is 3 seconds.	The response timeout for the RADIUS, TACACS+, and LDAP server.
base64-string-type	string	0|4–maximum ([!-~]{4,})?	The 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.
certificate-application-type	enumeration	commoncert|https|radius|ldap|syslog|fcap|all|extension	The certificate type.
certificate-entity-type	enumeration	cert|ca-client|ca-server|csr|ca	The certificate entity type.
default-string-type	string	1–maximum [-~]*	The default string.
gen-certificate-entity-type	enumeration	cert|csr	The certificate entity type.
hash-algorithm-type	enumeration	md5|sha258|sha512	The hash type (such as md5, sha258, or sha512)
home-virtual-fabric-type	fibrechannel:fabric-id-type	N/A	The account's home Virtual Fabric.
host-type	union	Any	The IP address. This data type is composed of the following data types: • inet:host • inet:ip-prefix • string
ipfilter-action-type	enumeration	activate|clone	The action to take on the IP filter policy.
ipfilter-ip-version-type	enumeration	IPv4|IPv6	The IP filter policy version (such as IPv4 or IPv6).
ipfilter-name-type	string	1–20 ([a-zA-Z0-9_]{1,20}) default_ipv4|default_ipv6 modifier = invert-match	The name of the IP filter policy.
ipfilter-permission-type	enumeration	permit|deny	The permit or deny action associated with this rule.
ipfilter-protocol-type	enumeration	tcp|udp	The protocol type (such as tcp or udp).
ipfilter-traffic-type	enumeration	INPUT|FORWARD	The type of traffic allowed for the specified IP address.

Name	Type	Range/Pattern/Length	Description
keysize-type	enumeration	1024|2048|4096|8192|P384	The size of the key in bytes. The greater the value, the more secure the connection; however, performance degrades with size.
ldap-tls-mode-type	enumeration	starttls|ldaps	The LDAP Transport Layer Security (TLS) mode for the LDAP server.
password-cfg-operation-type	enumeration	hash-config|default|delete-all	The configuration or deletion of password policies.
seccertmgmt-hash-type	enumeration	sha1|sha256|sha512	The hash type.
seccertmgmt-operation-type	enumeration	import|export	The certificate management operations.
seccertmgmt-protocol-type	enumeration	scp|ftp	The file transfer protocol.
sec-crypto-cfg-actions-type	enumeration	apply|verify|import|export	The action to perform on a template.
sec-crypto-cfg-application-type	enumeration	ssh|https	The application type.
sec-crypto-cfg-default-template-name-type	enumeration	default_generic|default_strong|default_fips|default_cc	The template's default name.
sec-crypto-cfg-file-transfer-protocol-type	enumeration	scp|sftp|ftp	The file transfer protocol.
sec-crypto-cfg-template-name-type	string	1–256 [a-zA-Z0-9_-]{1,256}	The template name.
sec-crypto-cfg-tls-cipher-type	string	1–64 [-+!0-9]{0,1}[a-zA-Z0-9]{2,64}	The cipher algorithms.
sshutil-algorithm-type	enumeration	rsa|dsa|ecdsa	The key algorithm type.
sshutil-hash-type	enumeration	md5|sha1|sha256|sha384|sha512	The hash type.
sshutil-key-type	enumeration	public-private-key|host-key	The sshutil key generation type.
sshutil-operation-type	enumeration	import|export	The sshutil operations.
tls-protocol-type	string	1–10 any|TLSv1.2|TLSv1|TLSv1.3	The TLS protocol.
user-config-access-hours-type	uint16	1–24 hours.	The hour.
user-config-access-minutes-type	uint16	1–60 minutes.	The minute.
user-config-role-type	string	4–16 [a-zA-Z]{4,16}	The account's role.
user-config-user-name-type	string	1–32 ([a-zA-Z][0-9a-zA-Z._]{0,31})	The user name. User names are case-sensitive and can contain up to 32 alphanumeric characters, including periods (.) and underscore (_) characters.
user-password-type	string	1–40 [-~]{1,40} : modifier = invert-match	Specifies a password for the account.

Name	Type	Range/Pattern/Length	Description
validation-mode-type	enumeration	basic|strict	The X509 validation mode.
virtual-fabric-action-type	enumeration	addlf|deletelf	The addition or deletion of Virtual Fabrics for user configuration.
virtual-fabric-role-id-type	string	N/A	The Virtual Fabrics to be added to the LDAP role. The format is *role=virtual-fabric-id-list* where *role* is the LDAP role and *virtual-fabric-id-list* is a list of comma-separated virtual fabrics.

SNMP Module Data Types

The following table describes additional SNMP module data types supported by the FOS REST API. Additional details about these data types can be found in the associated brocade-snmp-types.yang file. This file is located in the same folder as the module .yang files.

Table 15: SNMP Module Data Types

Name	Type	Range/Pattern/Length	Description
access-permission	enumeration	ro|rw	The SNMP access control permission.
authentication-protocol-type	enumeration	md5|sha|noauth	The authorization protocol for the SNMPv3 user.
default-control-type	enumeration	snmpv1|snmpv3|accesscontrol|systemgroup|mibcap|auditinterval|mibcapability	The SNMP default configuration options.
group-name	enumeration	ro|rw|read-only|read-write	The SNMP group name.
mibs-name	enumeration	FE-MIB|SW-MIB|FA-MIB|FICON-MIB|HA-MIB|FCIP-MIB|IF-MIB|BROCADE-MAPS-MIB|T11-FC-ZONE-SERVER-MIB	A list of supported SNMP MIBs.
privacy-protocol-type	enumeration	des|aes128|aes256|nopriv	The privacy protocol for the SNMPv3 user.
security-level	uint16	0..3	The SNMP security access level for the GET or SET operation.
severity-level	enumeration	none|critical|error|warning| informational| debug	The event notification severity level.

Name	Type	Range/Pattern/Length	Description
traps-name	enumeration	swFCPortScn|swEventTrap| swIPv6ChangeTrap| swPmgrEventTrap| swFabricReconfigTrap| swFabricSegmentTrap| swExtTrap|swStateChangeTrap| wPortMoveTrap| swBrcdGenericTrap| swDeviceStatusTrap| swZoneConfigChangeTrap| connUnitStatusChange| connUnitEventTrap| connUnitPortStatusChange| linkRNIDDeviceRegistration| linkRNIDDeviceDeRegistration| linkLIRRListenerRemoved| linkRLIRFailureIncident| fruStatusChanged| cpStatusChanged|fruHistoryTrap| linkDown|linkUp|mapsTrapAM| mapsQuietTimeExpirationTrap| t11ZsRequestRejectNotify| t11ZsMergeFailureNotify| t11ZsMergeSuccessNotify| t11ZsDefZoneChangeNotify| t11ZsActivateNotify	A list of supported SNMP traps.

Time Module Data Types

The following table describes additional time module data types supported by the FOS REST API. Additional details about these data types can be found in the associated brocade-time-types.yang file. This file is located in the same folder as the module .yang files.

Table 16: Time Module Data Types

Name	Type	Range/Pattern/Length	Description
brocade-time-type:ts-timezone-type	enumeration	N/A	The region and city of the time zone. for a complete list, refer to the brocade-time-types.yang file.
brocade-time-type:ts-ntp-type	union	1–32 LOCL	The NTP server IP address(es), hostname(s), or LOCL (for the local server).

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 Use Cases section of this publication or the RESTCONF RFC (RFC 8040).

The FOS REST API modules supported in Fabric OS 9.0.x are described in the following tables. All support the OPTIONS method.

Table 17: 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 18: FOS REST API Modules for Brocade Operations Features

Module Name	Description	Supported Methods
brocade-operation-device-management	Allows you to test the reachability of an HBA device.	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-fabric	Allows you to initiate a fabric build.	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-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-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-zone	Allows you to perform zone-related operations. You can also perform a zone expunge on zone objects.	POST, OPTIONS

Table 19: 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-chassis	Allows you to retrieve a detailed view of configuration and runtime information of the chassis.	GET, PATCH, HEAD, OPTIONS

Module Name	Description	Supported Methods
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-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/fibrechannel	Allows you to configure ports and retrieve port configurations.	GET, PATCH, 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-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

Module Name	Description	Supported Methods
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-zone	A data model for configuring zones and retrieving operational zoning information.	GET, POST, PATCH, DELETE, HEAD, OPTIONS

Table 20: FOS REST API Modules for Brocade Extension Features

Module Name	Description	Supported Methods
brocade-extension	Allows you to view information about Brocade extension objects as well as configure traffic control lists.	GET, POST, PATCH, HEAD, OPTIONS
brocade-extension-ip-interface	Allows you to manage the Brocade Extension IP interface.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
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-interface/portchannel	Allows you to view and configure port channel interfaces on the switch.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
brocade-extension-tunnel	Allows you to manage Brocade extension tunnels.	GET, POST, PATCH, DELETE, HEAD, OPTIONS
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

NOTE
This document does not address specific pattern or keyword requirements for individual leafs. Refer to the individual .yang files for explicit details on these aspects of the modules.

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-operation-show-status

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

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
```

URI Format

The URI format for this module takes the following form:

<base_URI>/operations/show-status/message-id/*message-id* followed by the leafs as listed in module tree to obtain the status of any REST operation (where *message-id* is the message ID generated when you initiated the REST operation).

Supported Methods

Only the POST operation is supported in this module.

History

Release Version	History
Fabric OS 8.2.1	This API call was introduced.
Fabric OS 9.0.0	Added the firmware download status and EULA text parameters.

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-supportsave

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

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
```

URI Format

The URI format for this module takes the following form:

<base_URI>/operations/supportsave to initiate supportSave.

Supported Methods

Only the OPTIONS and POST operations are supported in this module.

History

Release Version	History
Fabric OS 8.2.1	This API call was introduced.
Fabric OS 8.2.2	Added the port parameter to set the SCP or SFTP port number.
Fabric OS 9.0.0	Added the firmware download status, serial-mode, and EULA text parameters.

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>
```

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 of the Access Gateway (AG).

It assumes a knowledge of Access Gateway as performed in Fabric OS. For information on that topic, refer to the *Brocade Fabric OS Access Gateway Administration Guide*.

NOTE
The brocade-access-gateway module is supported in Fabric OS 8.2.0a and later.

Module Tree

This is the tree view of the module from the `brocade-ag.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]
       |  +--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
       |     +--rw load-balancing-mode-enabled?                   uint8
       |     +--rw multiple-fabric-name-monitoring-mode-enabled?  uint8
       +--rw n-port-map* [n-port]
       |  +--rw n-port                                            fibrechannel:slot-port-name-type
       |  +--rw failover-enabled?                                 uint8
       |  +--rw failback-enabled?                                 uint8
       |  +--ro online-status?                                    uint8
       |  +--ro reliable-status?                                  int8
       |  +--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
       +--ro f-port-list* [f-port]
       |  +--ro f-port                                            fibrechannel:slot-port-name-type
       |  +--ro online-status?                                    uint8
       |  +--ro f-port-info
       |     +--ro n-port?                                        fibrechannel:slot-port-name-type
       |     +--ro login-exceeded?                                uint8
       +--rw policy
       |  +--rw port-group-policy-enabled?                        uint8
       |  +--rw auto-policy-enabled?                              uint8
       +--rw n-port-settings
```

```
|  +--rw reliability-counter?                    yang:zero-based-counter64
+--ro device-list* [wwn]
   +--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
```

URI Format

The URI format for this module takes the following form:

• *<base_URI>*/running/brocade-access-gateway/ followed by the leafs as listed in the module tree.

• *<base_URI>*/running/brocade-access-gateway/n-port-map to configure N_Port mapping.

• *<base_URI>*/running/brocade-access-gateway/port-group to configure port groups.

• *<base_URI>*/running/brocade-access-gateway/f-port-list to view F_Port information.

• *<base_URI>*/running/brocade-access-gateway/policy to enable port group or auto policy.

• *<base_URI>*/running/brocade-access-gateway/n-port-settings to configure the reliability counter.

• *<base_URI>*/running/brocade-access-gateway/device-list to view a list of devices logged on to the Access Gateway switch and the device mapping with the F_Port, N_Port, and Fibre Channel ID (FCID).

• *<base_URI>*/running/brocade-interface/fibrechannel/name/*fibre-channel-interface-name*/n-port-enabled to configure a port to operate as an N_Port.

• *<base_URI>*/running/brocade-switch/fibrechannel-switch/name/*switch-worldwide-name*/ag-mode to determine if the switch is in Access Gateway mode or to configure the Access Gateway mode on the switch.

Parameters

brocade-access-gateway

Description: Configuration and runtime information of the Access Gateway.
Flag: read-write
This container has the following leafs:

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. This parameter is available only when the Port Grouping policy is enabled (port-group-policy-enabled = 1).
Flag: read-write
Key: *port-group-id*
This list has the following leafs:

port-group-id

Description: 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.
Flag: read-write
Type: uint8
Value: 0 to the maximum number of ports available on the given platform.
Optional: Yes

port-group-name

Description: The port group name.
Flag: read-write

Type: string
Value: 1 to 64 alphanumeric characters. The name must be alphanumeric. The default port group name uses the format pg<port-group-id>.
Optional: Yes

port-group-n-ports
Description: List of N_Ports that are in the port group.
Flag: read-write
This container has the following leaf:

n-port
Description: List of N_Ports that are in the port group. The port group must contain at least one N_Port.
Flag: read-write
Type: fibrechannel:slot-port-name-type
Value: The slot and port number of the specified port in the format slot/port.
Optional: Yes

port-group-f-ports
Description: List of F_Ports that are in the port group. This parameter is available only when load-balancing mode is enabled (load-balancing-mode-enabled = 1) for the port group.
Flag: read-write
This container has the following leaf:

f-port
Description: List of F_Ports that are in the port group. To update the configuration, you must enable load-balancing mode for the port group.
Flag: read-write
Type: fibrechannel:slot-port-name-type
Value: The slot and port number of the specified port in the format slot/port.
Optional: Yes

port-group-mode
Description: The mode configured for the port group.
Flag: read-write
This container has the following leafs:

load-balancing-mode-enabled
Description: Enables or disables load-balancing mode for the specified port group.
Flag: read-write
Type: uint8
Values: 0 = Load-balancing mode disabled. **1** = Load-balancing mode enabled. Default: 0.
Optional: Yes

multiple-fabric-name-monitoring-mode-enabled
Description: Enables or disables multiple fabric name monitor mode for the specified port group.
Flag: read-write
Type: uint8
Values: 0 = Multiple fabric name monitor mode disabled. **1** = Multiple fabric name monitor mode enabled. Default: 0.
Optional: Yes

n-port-map
Description: Defines the N_Port to F_Port mappings.
Flag: read-write

Key: *n-port*
This container has the following leafs:

n-port
Description: Enables mapping F_Ports to an N_Port.
Flag: read-write
Type: fibrechannel:slot-port-name-type
Value: The slot and port number of the specified port in the format slot/port.
Optional: Yes

failover-enabled
Description: Enables or disables failover for an N_Port.
Flag: read-write
Type: uint8
Values: 0 = Failover disabled. **1** = Failover enabled.
Optional: Yes

failback-enabled
Description: Enables or disables failback for an N_Port.
Flag: read-write
Type: uint8
Value: 0 = Failback disabled. **1** = Failback enabled.
Optional: Yes

online-status
Description: Whether the N_Port is online or offline.
Flag: read-only
Type: uint8
Config: false
Values: 0 = Offline. **1** = Online.
Optional: Yes

reliable-status
Description: The reliable status of the N_Port. This parameter is available only when the reliability counter is enabled (reliability-counter ! = 0).
Flag: read-only
Type: uint8
Config: false
Values: 0 = Unreliable. **1** = Reliable. **-1** = Not applicable.
Optional: Yes

n-port-info
Description: The N_Port logon information and the attached switch details. This parameter is available only when the port is online (online-status = 1).
Config: false
This container has the following leafs:

attached-fabric-name
Description: The WWN of the fabric attached to the N_Port.
Flag: read-only
Type: fibrechannel:wwn-type
Value: A valid WWN name.
Optional: Yes

attached-port-wwn
Description: The WWN of the port attached to the N_Port.
Flag: read-only

Type: fibrechannel:wwn-type
Value: A valid WWN name.
Optional: Yes

n-port-fcid
Description: The FCID of the N_Port.
Flag: read-only
Type: fibrechannel:fcid-type
Value: A valid destination FCID value.
Optional: Yes

attached-switch-user-friendly-name
Description: The ASCII name assigned to the switch by the administrator.
Flag: read-only
Type: string
Value: 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.
Optional: Yes

attached-switch-f-port
Description: The fabric switch port number of the port attached to the N_Port.
Flag: read-only
Type: fibrechannel:slot-port-name-type
Value: The slot and port number of the specified port in the format slot/port.
Optional: Yes

attached-switch-ip-address
Description: The out-of-band IP address of the attached switch.
Flag: read-only
Type: inet:ip-address
Value: A valid IPv4 or IPv6 address.
Optional: Yes

configured-f-port-list
Description: The F_Port to N_Port mappings.
Flag: read-write
This container has the following leaf:

f-port
Description: List of F_Ports that are mapped to the N_Port. There must be at least one F_Port mapped to an N_Port. Once you enable Access Gateway mode in the switch, F_Ports are mapped to the default N_Port. You must clear the existing mapping before you map an F_Port to another N_Port. Note that configured and static port mapping cannot overlap.
Flag: read-write
Type: fibrechannel:slot-port-name-type
Value: The slot and port number of the specified port in the format slot/port.
Optional: Yes

static-f-port-list
Description: The static F_Port to N_Port mappings.
Flag: read-write
This container has the following leaf:

f-port

Description: List of F_Ports that are statically mapped to the N_Port. There must be at least one F_Port statically mapped to an N_Port. Note that configured and static port mapping cannot overlap.
Flag: read-write
Type: fibrechannel:slot-port-name-type
Value: The slot and port number of the specified port in the format slot/port.
Optional: Yes

f-port-list

Description: List of all F_Ports present on the Access Gateway.
Flag: read-only
Key: *f-port*
Config: false
This container has the following leafs:

f-port

Description: The F_Port for which information is being fetched.
Flag: read-only
Type: fibrechannel:slot-port-name-type
Value: The slot and port number of the specified port in the format slot/port.
Optional: Yes

online-status

Description: Whether the F_Port is online or offline.
Flag: read-only
Type: uint8
Config: false
Values: 0 = Offline. **1** = Online.
Optional: Yes

f-port-info

Description: The F_Port login information. This parameter is available only when the port is online (online-status = 1).
Flag: read-only
Config: false
This container has the following leafs:

n-port

Description: The N_Port to which this F_Port is mapped.
Flag: read-only
Type: fibrechannel:slot-port-name-type
Value: The slot and port number of the specified port in the format slot/port.
Optional: Yes

login-exceeded

Description: The login exceeded state of the F_Port.
Flag: read-only
Type: uint8
Values: 0 = Login limit not exceeded. **1** = Login limit exceeded.
Optional: Yes

policy

Description: The Access Gateway policy configuration. To enable or disable any of the AG policies, the switch must be in a disabled state.
Flag: read-write
This container has the following leafs:

port-group-policy-enabled
Description: Enables or disables the port group policy. Note that the auto policy must be disabled (auto-policy-enabled = 0). To modify the policy configuration, the switch must be in a disabled state.
Flag: read-write
Type: uint8
Values: 0 = Disabled. **1** = Enabled.
Optional: Yes

auto-policy-enabled
Description: Enables or disables the auto policy. Note that the port group policy must be disabled (port-group-policy-enabled = 0). To modify the policy configuration, the switch must be in a disabled state.
Flag: read-write
Type: uint8
Values: 0 = Disabled. **1** = Enabled.
Optional: Yes

n-port-settings
Description: The N_Port-related configuration parameters. Note that the port group policy must be enabled (port-group-policy-enabled = 1).
Flag: read-write
This container has the following leaf:

reliability-counter
Description: The reliability counter configuration for N_Ports.
Flag: read-write
Type: yang:zero-based-counter64
Values: 0 = Disabled. **10**–**100** Enabled.
Optional: Yes

device-list
Description: A list of devices logged on to the Access Gateway and their F_Port, N_Port, and FCID mapping.
Key: wwn
Flag: read-only
Config: false
This container has the following leafs:

wwn
Description: The port world wide name of the connected device.
Flag: read-only
Type: fibrechannel:wwn-type
Values: A valid port WWN name.
Optional: Yes

fcid
Description: The fibre channel ID (FCID) of the connected device.
Flag: read-only
Type: fibrechannel:fcid-hex-string-type
Values: A vaild FCID in hexadecimal format.
Optional: Yes

f-port
Description: The F_Port to which the device is connected.
Flag: read-only
Type: fibrechannel:slot-port-name-type
Values: The slot and port number of the specified port in the format slot/port.
Optional: Yes

n-port

Description: The N_Port through which the device logged on.
Flag: read-only
Type: fibrechannel:slot-port-name-type
Values: The slot and port number of the specified port in the format slot/port.
Optional: Yes

Supported Methods

Only the OPTIONS, GET, PATCH, DELETE, HEAD, and POST operations are supported in this module.

Examples

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.

Viewing All Port Groups on the Access Gateway

The following example uses the GET request to retrieve the details for all port groups on the Access Gateway (AG).

Structure

GET *<base_URI>*/running/brocade-access-gateway/port-group

URI

```
GET https://10.10.10.10/rest/running/brocade-access-gateway/port-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 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-mode-
enabled>
        </port-group-mode>
    </port-group>
</Response>
```

Viewing Port Information

The following example uses the GET request to retrieve the port mapping information for N_Ports and F_Ports in the Access Gateway.

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>0/16</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/>
    </n-port-map>
     <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/>
    </n-port-map>
    <n-port-map>
        <n-port>0/18</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>
            <f-port>0/0</f-port>
            <f-port>0/2</f-port>
            <f-port>0/3</f-port>
        </configured-f-port-list>
        <static-f-port-list/>
    </n-port-map>
</Response>
```


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/>
    </n-port-map>
    <n-port-map>
        <n-port>0/18</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>
            <f-port>0/0</f-port>
            <f-port>0/2</f-port>
            <f-port>0/3</f-port>
        </configured-f-port-list>
        <static-f-port-list/>
    </n-port-map>
    <n-port-map>
        <n-port>0/19</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>
            <f-port>0/4</f-port>
            <f-port>0/5</f-port>
            <f-port>0/6</f-port>
            <f-port>0/7</f-port>
        </configured-f-port-list>
        <static-f-port-list/>
    </n-port-map>
    <n-port-map>
        <n-port>0/20</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>20:1b:c4:f5:7c:c0:fd:5d</attached-port-wwn>
            <n-port-fcid>0xeb1b00</n-port-fcid>
            <attached-switch-user-friendly-name>G620_066_235</attached-switch-user-friendly-name>
            <attached-switch-f-port>0/27</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/>
    </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/>
    </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>
```

History

Release Version	History
Fabric OS 8.2.0a	This API call was introduced.
Fabric OS 8.2.1	This API was modified to add the device-list parameter to the module.

brocade-chassis

This module provides a detailed view of configuration and runtime information of the Fabric OS switch or director.

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?   string
     |  +--ro chassis-wwn?                  fibrechannel:wwn-type
     |  +--ro serial-number?                fru:serial-number-type
     |  +--ro manufacturer?                 fru:manufacturer-type
     |  +--ro part-number?                  fru:part-number-type
     |  +--ro entitlement-serial-number?    fru:serial-number-type
     |  +--ro max-blades-supported?         uint16
     |  +--ro vendor-serial-number?         string
     |  +--ro vendor-part-number?           string
     |  +--ro vendor-revision-number?       string
     |  +--ro product-name?                 string
     |  +--rw vf-enabled?                   boolean {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro vf-supported?                 boolean {fibrechannel:fibrechannel_switch_platform}?
     |  +--ro date?                         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
```

URI Format

The URI format for this module takes the following form:

• *<base_URI>*`/running/brocade-chassis/chassis` followed by the leafs as listed in the module tree to retrieve information about the chassis.

• *<base_URI>*`/running/brocade-chassis/ha-status` followed by the leafs as listed in the module tree to retrieve the detailed High Availability (HA) status.

Parameters

brocade-chassis

Description: A detailed view of configuration and runtime information of the chassis.
Flag: read-write
This container has the following leafs:

chassis

Description: The complete details of the chassis.
Flag: read-write
Key: <*key*>
This list has the following leafs:

chassis-user-friendly-name

Description: 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, hyphens (-), and underscores (_). The name is not case-sensitive.
NOTE
In FICON mode, the name is limited to 1 to 24 alphanumeric characters plus hyphens (-) and underscores (_).
Flag: read-write
Type: string
Value: 1 to 31 alphanumeric characters plus hyphens (-) and underscores (_). A chassis name must begin with a letter. Spaces are not allowed.
Optional: Yes

chassis-wwn

Description: The WWN of the chassis, which is used for the license.
Flag: read-only
Type: fibrechannel:wwn-type
Config: false
Value: A vaild WWN.
Optional: Yes

serial-number

Description: A printable ASCII string that specifies the serial number of the chassis.
Flag: read-only
Type: fru:serial-number-type
Config: false
Value: 1 to 12 printable ASCII characters.
Optional: Yes

manufacturer

Description: The manufacturer of the chassis.
Flag: read-only
Type: fru:manufacturer-type
Config: false
Value: 1 to 63 characters.
Optional: Yes

part-number

Description: The part number for the physical element assigned by the manufacturer.
Flag: read-only
Type: fru:part-number-type
Config: false
Value: 1 to 14 characters.
Optional: Yes

entitlement-serial-number

Description: A serial number that is used for entitlement support. For Gen 5 chassis, the `entitlement-serial-number` leaf has the same output as the WWN 1 Factory Serial Number from the `chasissshow` command. For Gen 6 chassis, the `entitlement-serial-`

`number` leaf has the same output as the `serial-number` leaf (Chassis Factory Serial Number from the `chasissshow` command).
Flag: read-only
Type: fru:serial-number-type
Config: false
Value: A printable ASCII string that specifies the serial number of the chassis.
Optional: Yes

max-blades-supported
Description: The maximum number of blades that can fit in the physical chassis. This includes switch, control processor, application, and core routing blades.
Flag: read-only
Type: uint16
Config: false
Value: 1 = 1 blade (fixed-port switch). **8** = 8 blades. **12** = 12 blades.
Optional: Yes

vendor-serial-number
Description: The serial number of the chassis assigned by the vendor.
Flag: read-only
Type: string
Config: false
Value: 1 to 20 characters.
Optional: Yes

vendor-part-number
Description: The part number of the chassis assigned by the vendor.
Flag: read-only
Type: string
Config: false
Value: 1 to 20 characters.
Optional: Yes

vendor-revision-number
Description: The revision number of the chassis assigned by the vendor.
Flag: read-only
Type: string
Config: false
Value: 1 to 4 characters.
Optional: Yes

product-name
Description: The product name of the chassis.
Flag: read-only
Type: string
Config: false
Value: 1 to 255 characters.
Optional: Yes

vf-enabled
Description: Whether Virtual Fabrics is enabled on the chassis.
NOTE
Virtual Fabrics requires that Fibre Channel switch native mode be supported.
NOTE
Enabling Virtual Fabrics is a disruptive operation that requires a reboot to take effect.
Flag: read-write

Type: boolean
Value: true = Virtual Fabrics is enabled on the chassis. **false** = Virtual Fabrics is not enabled on the chassis.
Optional: Yes

vf-supported
Description: Whether Virtual Fabrics is supported on the chassis.
NOTE
Virtual Fabrics requires that Fibre Channel switch native mode be supported.
Flag: read-only
Type: boolean
Config: false
Value: true = Virtual Fabrics is supported on the chassis. **false** = Virtual Fabrics is not supported on the chassis.
Optional: Yes

date
Description: The current date of the switch.
Flag: read-only
Type: string
Config: false
Value: standard IETF date-time output
Optional: Yes

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 standby CP
• HA synchronization status
Flag: read-only
Config: false
This container has the following leafs:

active-cp
Description: The ID of the active CP.
Flag: read-only
Type: string
Value: cp0 or cp1
Optional: Yes

standby-cp
Description: The ID of the standby CP.
Flag: read-only
Type: string
Value: cp0 or cp1
Optional: Yes

active-slot
Description: The slot number of the active CP.
Flag: read-only
Type: uint16
Value: 0 through 12
Optional: Yes

standby-slot
Description: The slot number of the standby CP.
Flag: read-only
Type: uint16
Value: 0 through 12
Optional: Yes

recovery-type
Description: The recovery status of the switch.
Flag: read-only
Type: string
Value: cold or warm
Optional: Yes

standby-health
Description: The health status of the standby CP.
Flag: read-only
Type: string
Value: 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.
Optional: Yes

ha-enabled
Description: Whether High Availability (HA) is enabled or disabled.
Flag: read-only
Type: boolean
Value: true = HA is enabled. **false** = HA is disabled.
Optional: Yes

heartbeat-up
Description: Whether the heartbeat to the standby CP is up or down.
Flag: read-only
Type: boolean
Value: true = The heartbeat to the standby CP is up. **false** = The heartbeat to the standby CP is down.
Optional: Yes

ha-synchronized
Description: Whether HA is in a synchronized state. When HA is in a synchronized state and a failover becomes necessary, it is nondisruptive.
Flag: read-only
Type: boolean
Value: true = HA is synchronized. **false** = HA is not synchronized.
Optional: Yes

Supported Methods

Only the GET, PATCH, HEAD, and OPTIONS operations are supported in this module.

Examples

Viewing the Chassis Data

The following example uses the GET request to retrieve information about the chassis.

Structure

<base_URI>/running/brocade-chassis/chassis

URI

```
GET https://10.10.10.10/rest/running/brocade-chassis/chassis
```

Request Body

No request body is required.

Response Body

For Gen 5 chassis, the `entitlement-serial-number` leaf has the same output as the WWN 1 Factory Serial Number from the `chasissshow` command.

```
<?xml version="1.0"?>
<Response>
    <chassis>
        <chassis-user-friendly-name>b02-8514-60</chassis-user-friendly-name>
        <chassis-wwn>10:00:00:05:1e:68:cc:a7</chassis-wwn>
        <serial-number>ANP1919E007</serial-number>
        <entitlement-serial-number>ANN1919E001</entitlement-serial-number>
        <manufacturer>Brocade Communications Systems LLC</manufacturer>
        <part-number>60-1000888-05</part-number>
        <vf-enabled>true</vf-enabled>
        <vf-supported>true</vf-supported>
        <max-blades-supported>8</max-blades-supported>
        <vendor-revision-number/>
        <vendor-part-number>SLKWRM0000X4S</vendor-part-number>
        <vendor-serial-number/>
        <product-name>dcx8510-4</product-name>
        <date>12/01/2021-16:05:36</date>
    </chassis>
</Response>
```

For Gen 6 chassis, the `entitlement-serial-number` leaf has the same output as the `serial-number` leaf (Chassis Factory Serial Number from the `chasissshow` command).

```
{
    "Response": {
        "chassis": {
            "chassis-user-friendly-name": "b02-gx68-74",
            "chassis-wwn": "10:00:88:94:71:0c:b4:32",
            "serial-number": "EZA1010P009",
            "entitlement-serial-number": "EZA1010P009",
            "manufacturer": "Brocade Communications Systems LLC",
            "part-number": "60-1003194-02",
            "vf-enabled": true,
            "vf-supported": true,
```

```
            "max-blades-supported": 12,
            "vendor-revision-number": "",
            "vendor-part-number": "SLKWRM0000X68",
            "vendor-serial-number": "",
            "product-name": "x6-8",
            "date": "11/29/2021-11:22:18"
        }
    }
}
```

Viewing the Chassis Name

The following example uses the GET request to view the administrator-assigned name of the chassis.

Structure

<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>Brocade6510</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

<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>10:00:10:eb:1a:10:1e:78</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>Cold Recovery</recovery-type>
        <recovery-complete>true</recovery-complete> <<<<<<<
        <standby-health>Non-Redundant</standby-health>
        <ha-enabled>true</ha-enabled>
        <heartbeat-up>false</heartbeat-up>
        <ha-synchronized>false</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

<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

<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>
```

History

Release Version	History
Fabric OS 8.2.1	This API call was introduced.
Fabric OS 8.2.3b	This API call was modified to add the entitlement-serial-number leaf to the chassis container.

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.

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]
          +--ro name                          fibrechannel:wwn-type
          +--ro switch-user-friendly-name?    string
          +--ro chassis-wwn?                  fibrechannel:wwn-type
          +--ro chassis-user-friendly-name?   string
          +--ro domain-id?                    fibrechannel:domain-id-type
          +--ro principal?                    uint8
          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
          +--ro path-count?                   uint32
```

URI Format

The URI format for this module takes one of the following forms:

• *<base_URI>*`running/brocade-fabric/fabric-switch/` followed by the leafs as listed in the module tree to retrieve information on all the switches in a fabric.

• *<base_URI>*`running/brocade-fabric/fabric-switch/name/`*name* followed by the leafs as listed in the module tree to retrieve information on a specific switch.

Parameters

NOTE
The top-level container name changed from "fabric" to "brocade-fabric". The previous top-level container name "fabric" is still supported in this release.

brocade-fabric

Description: Fabric state parameters. Requests are made using queries specifying the vf-id of the specific fabric.
Flag: read-only
Optional: No
This container has the following leafs:

fabric-switch

Description: The list of configured switches in the fabric.
Flag: read-only
Optional: No
Key: *name*

This container has the following leafs:

name

Description: The Fibre Channel WWN of the switch.
Flag: read-only
Type: fibrechannel:wwn-type
Config: false
Value: A valid WWN name.
Optional: No

switch-user-friendly-name

Description: The ASCII name assigned to the switch by the administrator.
Flag: read-only
Type: string
Config: false
Value: 1 to 30 alphanumeric characters plus hyphens, periods, and underscore characters. Spaces are not allowed. A switch name can begin with either a letter or number, but a switch name that begins with a numeric (0-9) character must also have at least an underscore (_), hyphen (-), period (.) or alphabetic (A-Z, a-z) character. A switch name with only numeric characters is not valid.
Optional: Yes

chassis-wwn

Description: The chassis WWN.
Flag: read-only
Type: fibrechannel:wwn-type
Config: false
Value: A valid WWN name.
Optional: Yes

chassis-user-friendly-name

Description: The ASCII name assigned to the switch chassis by the administrator.
Flag: read-only
Type: string
Config: false
Value: 1 to 31 alphanumeric characters plus hyphens (-), periods (.), and underscore (_) characters. A chassis name must begin with a letter. Spaces are not allowed.
Optional: Yes

domain-id

Description: The highest level in the three-level addressing hierarchy used in the Fibre Channel address identifier. A domain typically is associated with a single Fibre Channel switch.
Flag: read-only
Type: fibrechannel:domain-id-type
Config: false
Value: 1 through 239. Default: 1
Optional: Yes

principal

Description: Indicates if this switch is the fabric's principal switch.
Flag: read-only
Type: uint8
Config: false
Value: 1 = This is the fabric's principal switch. 0 = This is not the fabric's principal switch.
Optional: Yes

fcid

Description: This parameter is deprecated. Use the fcid-hex parameter. The destination Fibre Channel ID (D_ID) of the switch (decimal format).
Flag: read-only
Type: fibrechannel:fcid-type
Config: false
Value: A valid destination FCID value.
Optional: Yes

fcid-hex

Description: The destination Fibre Channel ID (D_ID) of the switch (hexadecimal format).
Flag: read-only
Type: fibrechannel:fcid-hex-string-type
Config: false
Value: A valid destination FCID value.
Optional: Yes

ip-address

Description: The IPv4 address for the switch.
Flag: read-only
Type: inet:ipv4-address
Config: false
Value: A valid IPv4 address.
Optional: Yes

fcip-address

Description: The IPv4 address the switch is using for Fibre Channel over IP.
Flag: read-only
Type: inet:ipv4-address
Config: false
Value: A valid IPv4 address.
Optional: Yes

ipv6-address

Description: The IPv6 address for the switch.
Flag: read-only
Type: inet:ipv6-address
Config: false
Value: A valid IPv6 address.
Optional: Yes

firmware-version

Description: A human-readable string identifying the firmware version running on the switch.
Flag: read-only
Type: string
Config: false
Value: A Brocade-defined 11-character string identifying the software release installed on the switch.
Optional: Yes

path-count

Description: The number of paths available to each remote domain.
Flag: read-only
Type: uint32
Value: 0 to 16.
Optional: Yes

Supported Methods

Only the GET, OPTIONS, and HEAD operations are supported in this module.

Examples

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>10:10:14:15:1c: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>
      <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: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>
      <path-count>1</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>
```

History

Release version	History
Fabric OS 8.2.0	This API call was introduced.
Fabric OS 8.2.1	Refined the switch-user-friendly-name and chassis-user-friendly-name regular expressions. Increased the firmware-version length.
Fabric OS 8.2.1b	The top-level container name changed from "fabric" to "brocade-fabric". The previous top-level container name "fabric" is still supported in this release. Added the path-count parameter.

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.

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*.

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 Types for 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 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
          +--ro supported-speed?                fibrechannel:speed-type
```

```
+--ro current-port-speed?              fibrechannel: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
```

URI Format

The URI format for this module takes the following form:

- *<base_URI>*/running/brocade-fdmi/hba to display the FDMI information for the HBA.
- *<base_URI>*/running/brocade-fdmi/port to display all HBA ports in the FDMI database.

Parameters

brocade-fdmi

Description: A detailed view of the Fabric Device Management Interface (FDMI).
Flag: read-only
Config: false
This container has the following leafs:

hba

Description: List of HBA attributes registered with FDMI.
Key: *hba-id*
Flag: read-only
This list has the following leafs:

hba-id

Description: 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 (for example, across power cycles) as the HBA identifier for the HBA.
Flag: read-only
Type: fibrechannel:wwn-type

Value: A valid WWN name.
Optional: No

domain-id
Description: The domain directly attached to the HBA.
Flag: read-only
Type: fibrechannel:domain-id-type
Value: 1 through 239. Default: 1
Optional: No

manufacturer
Description:
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.
Flag: read-only
Type: string
Value: 1 to 63 printable ASCII characters.
Optional: Yes

serial-number
Description: 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.
Flag: read-only
Type: string
Value: 1 to 63 printable ASCII characters.
Optional: Yes

model
Description: 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 attribute to 63 bytes.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

model-description
Description: 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.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

node-name
Description: 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 matches 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 is not returned for the host bus adapter.
Flag: read-only
Type: fibrechannel:wwn-type
Value: A valid WWN (colon-separated hexadecimal octets, most significant bit first, with leading zeros.)
Optional: Yes

node-symbolic-name

Description: The name registered with the Name Server as a Name Server Symbolic Node Name object, which 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 for the host bus adapter.
Flag: read-only
Type: string
Value: 4 to 255 printable ASCII characters.
Optional: Yes

hardware-version

Description: 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.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

driver-version

Description: A printable ASCII string that identifies the version level of the driver software that controls a host adapter. If a host bus adapter is concurrently under the control of multiple driver software modules with different versions, this resource may indicate the versions for more than one driver module.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

option-rom-version

Description: A printable ASCII string that identifies the Option ROM or BIOS version of a host adapter.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

firmware-version

Description: 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 attribute may indicate the versions for more than one firmware module.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

os-name-and-version

Description: A printable ASCII string that describes the type and version of the operating system that controls the host bus adapter.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

max-ct-payload

Description: A 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 attribute will not be returned.
Flag: read-only
Type: uint32
Value: 0 to the maximum size CT payload in 32-bit words.
Optional: Yes

vendor-id

Description: The T10 vendor ID of the manufacturer of the HBA, or an OEM of the HBA.
Flag: read-only
Type: string
Value: 1 to 8 printable ASCII characters.
Optional: Yes

vendor-specific-info

Description: A value with vendor-specific use.
Flag: read-only
Type: fibrechannel:brocade-hex-string-type
Value: 0 to 10 hexidecimal string.
Optional: Yes

number-of-ports

Description: The number of Nx_Ports on the HBA.
Flag: read-only
Type: uint32
Value: The number of Nx_Ports on the HBA.
Optional: Yes

fabric-name

Description: 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, fabric-name will not be provided.
Flag: read-only
Type: fibrechannel:wwn-type
Values: A valid WWN name.
Optional: Yes

boot-bios-version

Description: A printable ASCII string that describes the identification and version of a boot BIOS provided by the HBA.
Flag: read-only
Type: string
Value: 4 to 255 printable ASCII characters.
Optional: Yes

boot-bios-enabled

Description: Whether a boot BIOS provided by the HBA is enabled or disabled. boot-bios-enabled is true if the HBA provides a boot BIOS and the boot BIOS is enabled. boot-bios-enabled is false if the HBA provides a boot BIOS and the boot BIOS is disabled.
Flag: read-only
Type: uint8
Values: 1 = Boot-BIOS is enabled. 0 = Boot-BIOS is disabled.
Optional: Yes

hba-port-list

Description: A list of port WWNs associated with the HBA. This list corresponds with the port-name resource under the port list.
Flag: read-only
Optional: Yes
This list has the following leaf:

wwn

Description: The WWN of the port on the HBA. This corresponds to the port-name resource under the port list.
Flag: read-only
Type: fibrechannel:wwn-type
Values: A valid WWN name.
Optional: Yes

port

Description: A list of HBA port attributes registered with FDMI.
Flag: read-only
Key: port-name
This list has the following leafs:

port-name

Description: A WWN that identifies the port associated with the host bus adapter specified by the hba-id resource.
Flag: read-only
Type: fibrechannel:wwn-type
Value: A valid fibre channel WWN.
Optional: No

hba-id

Description: 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 (for example, across power cycles) as the HBA identifier for the HBA. This corresponds to the hba-id resource under the hba list.
Flag: read-only
Type: fibrechannel:wwn-type
Value: A valid fibre channel WWN.
Optional: No

domain-id

Description: The domain directly attached to the HBA port.
Flag: read-only
Type: fibrechannel:domain-id-type
Value: A FCID domain identifier.
Optional: No

port-symbolic-name

Description: The Port Symbolic Name object registered with the Name Server, which is subject to all the description and constraints of that object defined in FC-GS.
Flag: read-only
Type: string
Value: A valid port symbolic name.
Optional: Yes

port-id

Description: The Port Identifier object registered with the Name Server, which is subject to all the description and constraints of that object defined in FC-GS.
Flag: read-only
Type: fibrechannel:fcid-hex-string-type
Value: A vaild fibre channel ID (FCID) in hexadecimal format.
Optional: Yes

port-type

Description: The Name Server port type object registered with the Name Server, which is subject to all the description and constraints of that object defined in FC-GS.
Flag: read-only
Type: fibrechannel:port-type-string-type
Value: The port type.
n-port = (0x01)
nl-port = (0x02)
f/nl-port = (0x03)
nx-port = (0x7F)
f-port = (0x81)
fl-port = (0x82)
e-port = (0x84)
b-port = (0x85)
a-port = (0x86)
0xnn = (Unknown/Reserved <hexadecimal value>)
Optional: Yes

supported-class-of-service

Description: The Class of Service object registered with the Name Server, which is subject to all the description and constraints of that object defined in FC-GS.
Flag: read-only
Type: fibrechannel:class-of-service-type
Value: The class of service type.
class-f = (bit 0)
class-1 = (bit 1)
class-2 = (bit 2)
class-3 = (bit 3)
Optional: Yes

supported-fc4-type

Description: The FC-4 types attribute registered with the HBA Management Server as a port attribute. An Nx_Port registers a supported FC-4 types value that indicates "support" for any FC-4 type that it is able to be configured to support.
Flag: read-only
Type: fibrechannel:fc4-type-type
Value: The FC-4 type.
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)
Optional: Yes

active-fc4-types

Description: The Port active FC-4 types attribute registered with the Name Server, which is subject to all the description and constraints of that object defined in FC-GS. An Nx_Port registers a supported FC-4 types value that indicates "support" for any FC-4 type that it is able to be configured to support.
Flag: read-only
Type: fibrechannel:fc4-type-type
Value: The FC-4 type. The same values as the *supported-fc4-type*.
Optional: Yes

supported-speed

Description: The supported transmission speeds of the Nx_Port.
Flag: read-only
Type: fibrechannel:speed-type
Values:
speed-1-gfc = (Mask Value (hex): 0000 0001)
speed-2-gfc = (Mask Value (hex): 0000 0002)
speed-10-gfc = (Mask Value (hex): 0000 0004)
speed-4-gfc = (Mask Value (hex): 0000 0008)
speed-8-gfc = (Mask Value (hex): 0000 0010)
speed-16-gfc = (Mask Value (hex): 0000 0020)
speed-32-gfc = (Mask Value (hex): 0000 0040)
speed-20-gfc = (Mask Value (hex): 0000 0080)
speed-40-gfc = (Mask Value (hex): 0000 0100)
speed-128-gfc = (Mask Value (hex): 0000 0200)
speed-64-gfc = (Mask Value (hex): 0000 0400)
speed-256-gfc = (Mask Value (hex): 0000 0800)
speed-not-established = (Mask Value (hex): 0000 8000)
speed-10-ge = (Mask Value (hex): 0001 0000)
speed-40-ge = (Mask Value (hex): 0002 0000)
speed-100-ge = (Mask Value (hex): 0004 0000)
speed-25-ge = (Mask Value (hex): 0008 0000)
speed-50-ge = (Mask Value (hex): 0010 0000)
speed-400-ge = (Mask Value (hex): 0020 0000)
Optional: Yes

current-port-speed

Description: The Current Port Speed attribute returned by the HBA Management Server as a port attribute if either it has been registered or it has been determined by the HBA Management Server. If the HBA Management Server is able to determine a speed, the value returned for the Current Port Speed attribute indicates the speed determined by the HBA Management Server, regardless of any value registered.
Flag: read-only
Type: fibrechannel:speed-type

Value: The current port speed in gigabits per second (Gb/s). The same values as the *supported-speed*.
Optional: Yes

maximum-frame-size
Description: The Maximum Frame Size attribute registered with the HBA Management Server as a Port attribute. Maximum Frame Size attribute does not include the FC header but does include any optional headers.
Flag: read-only
Type: uint32
Value: The maximum FC frame payload in bytes.
Optional: Yes

os-device-name
Description: The OS Device Name attribute registered with the HBA Management Server as a port attribute. The OS Device Name attribute contains a printable ASCII character 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 attribute may be a comma-separated list of as many such names as fit in 255 bytes. If the software that registers Nx_Port attributes cannot determine any such OS Device Name, it does not register this attribute.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

host-name
Description: The Host Name attribute registered with the HBA Management Server as a Port attribute. If there are several such names that reference the same host, this attribute may be a comma-separated list of as many such names as fit in 255 bytes. If the software that registers Nx_Port attributes is unable to determine any such Host Name, it does not register this attribute.
Flag: read-only
Type: string
Value: The host associated with the Nx_Port. 1 to 255 printable ASCII characters.
Optional: Yes

node-name
Description: The Node Name attribute registered with the Name Server for the Nx_Port. The format of the Node_Name attribute is the format of the Name_Identifier described in FC-FS.
Flag: read-only
Type: fibrechannel:wwn-type
Value: A fibre channel WWN that identifies the node that contains the Nx_Port.
Optional: Yes

fabric-name
Description: The Port Fabric Name attribute registered with the HBA Management Server as an Nx_Port attribute. The Port Fabric Name attribute contains an 8-byte binary value equal to the Fabric_Name of the fabric associated with the Nx_Port (see FC-SW). Registration of the Port Fabric Name attribute is optional, and may be registered automatically by the fabric for Nx_Ports that are registered with the Fabric Name Server.
Flag: read-only
Type: fibrechannel:wwn-type
Value: A fibre channel WWN.
Optional: Yes

port-state

Description: 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 are vendor specific.
Flag: read-only
Type: fibrechannel:brocade-hex-string-type
Values:
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
Optional: Yes

number-of-discovered-ports

Description: 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 it may not reflect all nodes on a network.
Flag: read-only
Type: uint32
Value: The number of FC_Ports that are visible to the specified Nx_Port.
Optional: Yes

vsa-service-category

Description: The vendor-specific service category resource.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

vsa-guid

Description: The vendor-specific guid resource.
Flag: read-only
Type: string
Value: 1 to 34 printable ASCII characters.
Optional: Yes

vsa-version

Description: The vendor-specific version resource.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

vsa-product-name

Description: The vendor-specific product name resource.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

vsa-port-info
Description: The vendor-specific port information resource.
Flag: read-only
Type: string
Value: 1 to 63 printable ASCII characters.
Optional: Yes

vsa-qos-supported
Description: The vendor-specific QOS supported resource.
Flag: read-only
Type: string
Value: 1 to 63 printable ASCII characters.
Optional: Yes

vsa-security
Description: The vendor-specific security resource.
Flag: read-only
Type: string
Value: 1 to 63 printable ASCII characters.
Optional: Yes

vsa-storage-array-family
Description: The vendor-specific storage array family.
Flag: read-only
Type: string
Value: 1 to 63 printable ASCII characters.
Optional: Yes

vsa-storage-array-name
Description: The vendor-specific storage array name.
Flag: read-only
Type: string
Value: 1 to 63 printable ASCII characters.
Optional: Yes

vsa-storage-array-system-model
Description: The vendor-specific storage array system model.
Flag: read-only
Type: string
Value: 1 to 63 printable ASCII characters.
Optional: Yes

vsa-storage-array-os
Description: The vendor-specific storage array operating system.
Flag: read-only
Type: string
Value: 1 to 63 printable ASCII characters.
Optional: Yes

vsa-storage-array-node-count
Description: The vendor-specific storage array node count.
Flag: read-only
Type: uint32
Value: The storage array node count.
Optional: Yes

vsa-storage-array-nodes

Description: A list of vendor-specific storage array nodes.
Flag: read-only
Optional: Yes
This container has the following leaf.

*nodes**

Description: The vendor-specific storage array node resource.
Flag: read-only
Type: string
Value: 1 to 255 printable ASCII characters.
Optional: Yes

vsa-connected-ports

Description: A list of vendor-specific connected ports.
Flag: read-only
Optional: Yes
This container has the following leaf.

*wwns**

Description: The vendor-specific connected port resource.
Flag: read-only
Type: fibrechannel:wwn-type
Value: A valid fibre channel WWN.
Optional: Yes

Supported Methods

The GET, OPTIONS, and HEAD operations are supported in this module.

Examples

Retrieving FDMI Information From an HBA

This example uses a GET request to retrieve FDMI information for the HBA to which the request is sent.

Structure

GET *<base_URI>*/running/brocade-fdmi/hba

URI

```
GET https://10.10.10.10/rest/running/brocade-fdmi/hba
```

Request Body

The request body is empty.

Response Body

```
<?xml version="1.0"?>
<Response>
    <hba>
        <hba-id>21:00:00:0e:1e:1b:f9:10</hba-id>
        <domain-id>150</domain-id>
        <hba-port-list>
            <wwn>21:00:00:0e:1e:1b:f9:10</wwn>
        </hba-port-list>
        <node-name>20:00:00:0e:1e:1b:f9:10</node-name>
```

```
            <manufacturer>QLogic Corporation</manufacturer>
            <serial-number>BFE1425K14446</serial-number>
            <model>QLE2672</model>
            <model-description>QLE2672 QLogic 2-port 16Gb Fibre Channel Adapter</model-description>
            <hardware-version>HW VERSION</hardware-version>
            <driver-version>2.1.46.0</driver-version>
            <option-rom-version>3.40</option-rom-version>
            <firmware-version>8.03.06 (d0d5)</firmware-version>
            <os-name-and-version>VMware ESXi-6.5.0 (Releasebuild-4564106)</os-name-and-version>
            <max-ct-payload>2048</max-ct-payload>
            <node-symbolic-name>QLE2672 FW:v8.03.06 DVR:v2.1.46.0</node-symbolic-name>
            <vendor-specific-info>0x00001077</vendor-specific-info>
            <number-of-ports>1</number-of-ports>
            <fabric-name>20:00:00:0e:1e:1b:f9:10</fabric-name>
            <boot-bios-version>3.40</boot-bios-version>
            <boot-bios-enabled>1</boot-bios-enabled>
            <vendor-id>QLogic</vendor-id>
        </hba>
        <hba>
            <hba-id>21:00:00:0e:1e:18:99:71</hba-id>
            <domain-id>150</domain-id>
            <hba-port-list>
                <wwn>21:00:00:0e:1e:18:99:71</wwn>
            </hba-port-list>
            <node-name>20:00:00:0e:1e:18:99:71</node-name>
            <manufacturer>QLogic Corporation</manufacturer>
            <serial-number>BFE1346E15893</serial-number>
            <model>QLE2672</model>
            <model-description>QLE2672 QLogic 2-port 16Gb Fibre Channel Adapter</model-description>
            <hardware-version>HW VERSION</hardware-version>
            <driver-version>2.1.46.0</driver-version>
            <option-rom-version>3.40</option-rom-version>
            <firmware-version>8.03.06 (d0d5)</firmware-version>
            <os-name-and-version>VMware ESXi-6.5.0 (Releasebuild-4564106)</os-name-and-version>
            <max-ct-payload>2048</max-ct-payload>
            <node-symbolic-name>QLE2672 FW:v8.03.06 DVR:v2.1.46.0</node-symbolic-name>
            <vendor-specific-info>0x00001077</vendor-specific-info>
            <number-of-ports>1</number-of-ports>
            <fabric-name>20:00:00:0e:1e:18:99:71</fabric-name>
            <boot-bios-version>3.40</boot-bios-version>
            <boot-bios-enabled>1</boot-bios-enabled>
            <vendor-id>QLogic</vendor-id>
        </hba>
    </Response>
```

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>
        <hba-id>21:00:00:0e:1e:1b:f9:10</hba-id>
        <domain-id>150</domain-id>
        <port-name>21:00:00:0e:1e:1b:f9:10</port-name>
        <supported-fc4-type>FCP</supported-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>vmhba2</os-device-name>
        <host-name>537283a7-9d2d-a165-4c74-2c768a51</host-name>
        <node-name>20:00:00:0e:1e:1b:f9:10</node-name>
        <port-symbolic-name>QLE2672 FW:v8.03.06 DVR:v2.1.46.0 port</port-symbolic-name>
        <port-type>nx-port</port-type>
        <supported-class-of-service>class-3</supported-class-of-service>
        <fabric-name>20:a6:00:05:33:58:8d:00</fabric-name>
        <active-fc4-type>FCP</active-fc4-type>
        <port-state>0x2</port-state>
        <number-of-discovered-ports>1</number-of-discovered-ports>
        <port-id>0x96a640</port-id>
    </port>
    <port>
        <hba-id>21:00:00:0e:1e:18:99:71</hba-id>
        <domain-id>150</domain-id>
        <port-name>21:00:00:0e:1e:18:99:71</port-name>
        <supported-fc4-type>FCP</supported-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>vmhba3</os-device-name>
        <host-name>537283a7-c500-c440-edf6-2c768a51</host-name>
        <node-name>20:00:00:0e:1e:18:99:71</node-name>
        <port-symbolic-name>QLE2672 FW:v8.03.06 DVR:v2.1.46.0 port</port-symbolic-name>
        <port-type>nx-port</port-type>
        <supported-class-of-service>class-3</supported-class-of-service>
        <fabric-name>2e:2e:00:05:33:58:8d:00</fabric-name>
        <active-fc4-type>FCP</active-fc4-type>
        <port-state>0x2</port-state>
        <number-of-discovered-ports>1</number-of-discovered-ports>
        <port-id>0x96a6c0</port-id>
    </port>
</Response>
```

History

Release Version	History
Fabric OS 8.2.0a	This API call was introduced.

brocade-fibrechannel-configuration

This module enables you to configure a switch or director.

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 for data type descriptions.

```
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
      | +--rw area-mode?                    uint8 {fibrechannel:fibrechannel_chassis_platform}?
      +--rw f-port-login-settings
      | +--rw max-logins?                   uint16
      | +--rw max-flogi-rate-per-switch?    uint16
      | +--rw stage-interval?               uint16
      | +--rw free-fdisc?                  uint16
      | +--rw enforce-login?                uint16
      | +--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
      | +--rw node-name-zoning-enabled?     boolean {fibrechannel:fibrechannel_switch_platform}?
      +--rw fabric
         +--rw insistent-domain-id-enabled?  boolean {fibrechannel:fibrechannel_switch_platform}?
```

URI Format

The URI format for this module takes the following form:

- *<base_URI>*`/running/brocade-fibrechannel-configuration/switch-configuration` as followed by the leafs as listed in the module tree to view or configure the preconfigured parameters of the switch.
- *<base_URI>*`/running/brocade-fibrechannel-configuration/f-port-login-settings` as followed by the leafs as listed in the module tree to view or configure the F_Port login parameters.
- *<base_URI>*`/running/brocade-fibrechannel-configuration/port-configuration` as followed by the leafs as listed in the module tree to view or configure preconfigured port parameters.
- *<base_URI>*`/running/brocade-fibrechannel-configuration/zone-configuration` as followed by the leafs as listed in the module tree to view or configure zoning parameters.

Parameters

brocade-fibrechannel-configuration

Description: This module enables you to configure a switch or director.
Flag: read-write

This container has the following leafs:

switch-configuration

Description: Provides switch configuration parameters.
Flag: read-write
This container has the following leafs:

trunk-enabled

Description: Enables or disables trunking on all ports in the logical switch. To enable or disable trunking on individual ports, see the trunk-port-enabled leaf in the brocade-fibrechannel module. Note that this operation is persistent and disruptive for all ports in the logical switch. Note that while you can enable or disable trunking through a PATCH request you cannot retrieve information about this parameter in a GET request.
Flag: read-write
Type: boolean
Value: true = Trunking is enabled on all ports in the logical switch. **false** = Trunking is disabled on all ports in the logical switch.
Optional: Yes

wwn-port-id-mode

Description: Indicates whether WWN-based persistent PID is enabled on a switch. WWN port ID mode supports both dynamic and static WWN-based PID assignment. Every time the device logs on to the switch, the device is guaranteed the same PID. This parameter is not valid when the logical switch is configured as a base switch switch, and it cannot be set to true when the area-mode leaf is set to 2.
Flag: read-write
Type: uint8
Type: boolean
Value: true = WWN port ID mode is enabled on the switch. **false** = WWN port ID mode is disabled on the switch.
Optional: Yes

edge-hold-time

Description: The maximum time, in milleseconds, that a frame can wait after it is received on the ingress port and before it is delivered to the egress port.
Flag: read-write
Type: milliseconds
Value: For a default switch, valid values can be 80 to 500. Note that user-defined values are allowed only on a default switch. 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, Medium (500 ms).
Optional: Yes

area-mode

Description: The area mode. Area mode has a different meaning (see Value below) in a default switch and non-default, non-FICON switch. Note that the switch must be in disabled state for this setting (fibrechannel-switch:enabled-state = 3).
Flag: read-write
Type: uint8
Value: For a default switch, valid values include: **0** = 10-bit addressing mode (default); **1** = Zero-based area assignment (8-bit addressing mode).
For a nondefault and non-FICON logical switch, valid values include: **0** = 10-bit addressing mode (default); **1** = Zero-based area assignment; **2** = Port-based area assignment. A non-default logical switch with FICON mode enabled is not supported.
Optional: Yes

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 (fibrechannel-switch:enabled-state = 3).
Flag: read-write
This container has the following leafs:

max-logins

Description: The maximum number of allowed logins for the switch.
Flag: read-write
Type: uint16
Value: 0 to the maximum. The maximum value is determined by multiplying 126 by the number of ports in the switch (126 * *number_of_ports_in_switch*). The initial default value is 16 * *number_of_ports_in_switch*.
Optional: Yes

max-flogi-rate-per-switch

Description: The maximum number of fabric logins (FLOGIs) allowed, per second, on a switch.
Flag: read-write
Type: uint16
Value: 0 to the maximum.
Optional: Yes

stage-interval

Description: The interval setting, in milliseconds, for the rate at which F_Ports are enabled.
Flag: read-write
Type: uint16
Value: 0 to 10000 milliseconds.
Optional: Yes

free-fdisc

Description: The number of logins allowed before staging. 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.
Flag: read-write
Type: uint16
Value: 0 = Disables the staging of FDISC logins. **1** to **255** = Enables staging and allows the specified number of logins.
Optional: Yes

enforce-login

Description: The precedence for login when two devices with same the port WWN (PWWN) compete for login. All modes are for NPIV and non-NPIV F_Ports.
Flag: read-write
Type: uint16
Value: 0 = The first login takes precedence over the second login. **1** = The second login overrides the first login. **2** = For FDISC, the second FDISC login takes precedence. For FLOGI, the first FLOGI takes precedence.
Optional: Yes

max-flogi-rate-per-port

Description: The maximum number of logins allowed, per second, on a given port. If the number is exceeded, the port is fenced.
Flag: read-write
Type: uint16

Value: 0 to 100 logins per second.
Optional: Yes

port-configuration
Description: Provides the port configuration parameters.
Flag: read-write
This container has the following leafs:

portname-mode
Description: The current port name mode (**off**, **default**, **fdmi** or **dynamic**).
Flag: read-write
Type: string
Value: off = No port name configuration. **default** = The port name corresponds to the port index. **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. Note that **fdmi** and **dynamic** modes are not available in Access Gateway mode.
Optional: Yes

dynamic-portname-format
Description: 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 (note that I and C are mutually exclusive)
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. The switch must be an FC switch in Native mode.
Flag: read-write
Type: string
Value: 0 to 13 alphanumeric characters.
Optional: Yes

dynamic-d-port-enabled
Description: The port has D_Port capability, but it is not explicitly configured. Enabling a dynamic D_Port switch-wide configuration forces the ports on that switch or chassis to respond to D_Port requests from the other end of the connection. Basically, the port responds to a remote port request to change its mode to D_Port mode and run diagnostic tests automatically.
Flag: read-write
Type: boolean
Value: true = Dynamic D_Port configuration is enabled. **false** = Dynamic D_Port configuration is disabled.
Optional: Yes

on-demand-d-port-enabled
Description: The port has D_Port capability, but it is not explicitly configured. Enabling an on-demand D_Port switch-wide configuration forces the ports on that switch or chassis to respond to an internal request within the switch as a result of certain events (slot power off or on, persistent disable or enable, and so on). Basically, the switch responds to an internal request to change a port mode to D_Port mode, and run

diagnostic tests automatically. The D_Ports change to normal port mode after successful completion of the tests.
Flag: read-write
Type: boolean
Value: true = On-demand D_Port configuration is enabled. **false** = On-demand D_Port configuration is disabled.
Optional: Yes

zone-configuration

Description: Provides zoning configuration parameters.
Flag: read-write
This container has the following leaf:

node-name-zoning-enabled

Description: Indicates whether node name checking for zoning is enabled. The switch must be an FC switch in Native mode. The switch must be disabled to change this value.
Flag: read-write
Type: boolean
Value: true = Node name checking for zoning is enabled. **false** = Node name checking for zoning is disabled.
Optional: Yes

fabric

Description: Provides attributes applicable for a switch.
Flag: read-write
This container has the following leaf:

insistent-domain-id-enabled

Description: Indicates whether Insistent Domain ID (IDID) is enabled. Typically, the fabric automatically resolves domain ID conflicts during fabric merges or builds unless IDID is configured. If 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. The switch must be an FC switch in Native mode.
NOTE
For FICON switches, the insistent-domain-id-enabled leaf is read only.
Flag: read-write
Type: boolean
Value: true = IDID is enabled. **false** = IDID is disabled.
Optional: Yes

Supported Methods

Only the OPTIONS, GET, PATCH, and HEAD operations are supported in this module.

Examples

Viewing Switch Parameters

The following example uses the GET request to view existing parameters for a switch.

Structure

GET *<base_URI>*/running/brocade-fibrechannel-configuration/switch-configuration

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-configuration/switch-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>
    <switch-configuration>
        <area-mode>1</area-mode>
        <wwn-port-id-mode>false</wwn-port-id-mode>
        <edge-hold-time>500</edge-hold-time>
    </switch-configuration>
</Response>
```

Viewing F_Port Login Settings

The following example uses the GET request to view existing parameters for F_Port login.

Structure

GET *<base_URI>*/running/brocade-fibrechannel-configuration/f-port-login-settings

URI

```
GET https://10.10.10.10/rest/running/brocade-fibrechannel-configuration/f-port-login-settings
```

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>
    <f-port-login-settings>
        <max-logins>5120</max-logins>
        <max-flogi-rate-per-switch>0</max-flogi-rate-per-switch>
        <stage-interval>0</stage-interval>
        <free-fdisc>0</free-fdisc>
        <enforce-login>0</enforce-login>
        <max-flogi-rate-per-port>100</max-flogi-rate-per-port>
    </f-port-login-settings>
</Response>
```

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/dynamic-portname-format

URI

```
PATCH https://10.10.10.10/rest/running/brocade-switch/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/node-name-zoning-enabled

URI

```
PATCH https://10.10.10.10/rest/running/brocade-switch/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.

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-switch/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.

History

Release Version	History
Fabric OS 8.2.1	This API call was introduced.

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 information on diagnostics, refer to the *Brocade Fabric OS Troubleshooting and Diagnostics Guide*.

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                              string
        +--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
        +--ro electrical-loopback-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 optical-loopback-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 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
```

URI Format

The URI format for this module takes the following form:

<base_URI>/running/brocade-fibrechannel-diagnostics/fibrechannel-diagnostics/ followed by the leafs as listed in the module tree.

Supported Methods

The GET, HEAD, PATCH, and OPTIONS are supported in this module.

History

Release Version	History
Fabric OS 8.2.0	This API call was introduced.
Fabric OS 8.2.1b	The top-level container name changed from "diagnostics" to "brocade-fibrechannel-diagnostics". The previous top-level container name "diagnostics" is still supported in this release. However, new features (Fabric OS 9.0.0 or later) are not supported in the previous top-level container name "fabric".
Fabric OS 9.0.0	This API call was modified to add the remote-switch-wwn and remote-port-index parameters to the 'brocade-fibrechannel-diagnostics' container; add the start-time and estimate-time parameters to the electrical-loopback-test, optical-loopback-test, and link-traffic-test containers; add the failure-report-local-errors and failure-report-remote-errors containers to the failure-report container.
Fabric OS 9.1.0	This API call was modified to add the error-message leaf. This API call was modified to edit the start-time, end-time, result-type, distance, electrical-loopback-test, optical-loopback-test, and state leafs and the brocade-fibrechannel-diagnostics container.

brocade-fibrechannel-logical-switch

This module retrieves information of all logical switches in a operational chassis. This module also provides a logical switch configuration.

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 Administrator's 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 Types for data type descriptions.

```
module: brocade-fibrechannel-logical-switch
    +--rw brocade-fibrechannel-logical-switch
       +--rw fibrechannel-logical-switch* [fabric-id]
          +--rw fabric-id                uint32
          +--ro switch-wwn?              fibrechannel:wwn-type
          +--rw base-switch-enabled?     uint8
          +--ro default-switch-status?   uint8
          +--rw logical-isl-enabled?     uint8
          +--rw ficon-mode-enabled?      uint8
          +--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
```

URI Format

The URI format for this module takes the following form:

<base_URI>`/running/brocade-fibrechannel-logical-switch/fibrechannel-logical-switch` to retrieve logical switch information or configure a logical switch.

Parameters

NOTE
The top-level container name changed from "logical-switch" to "brocade-fibrechannel-logical-switch". The previous top-level container name "logical-switch" is still supported in this release.

brocade-fibrechannel-logical-switch

Description: The base container for this module.
Flag: read-write
This container has the following leafs:

fibrechannel-logical-switch

Description: Provides logical switch state parameters of all configured logical switches.
Flag: read-write
Key: fabric-id
This container has the following leafs:

fabric-id

Description: The virtual fabric identification (VFID) of the logical switch.
Flag: read-write
Type: uint32
Value: 1 to 128 alpha characters.
Optional: No

switch-wwn

Description: The switch WWN, which is a unique numeric identifier for the switch assigned by the chassis.
Flag: read-only
Type: fibrechannel:wwn-type
Value: A valid WWN for the switch.
Config: false
Optional: Yes

base-switch-enabled

Description: The logical switch that is the base switch in the chassis. You can configure the base switch when creating a logical switch. This parameter is available only when XISL and ficon are disabled.
Flag: read-write
Type: uint8
Value: 0 = Disabled. 1 = Enabled. Default is 0.
Optional: Yes

default-switch-status

Description: The logical switch is the default switch in the chassis.
Flag: read-only
Type: uint8
Value: 0 = Disabled. 1 = Enabled.
Config: false
Optional: Yes

logical-isl-enabled

Description: Enables logical ISLs (LISLs) on a logical switch. You can only disable LISLs when you create a logical switch. This parameter is available only when the base switch is disabled (base-switch-enabled=0).
Flag: read-write
Type: uint8
Value: 0 = Disabled. 1 = Enabled. Default is 1.
Optional: Yes

ficon-mode-enabled

Description: Indicates the FICON mode. This parameter is available only when the base switch is disabled (base-switch-enabled=0) and default switch status is disabled (default-switch-status=0). You cannot disable FICON mode in a logical switch.
Flag: read-write
Type: uint8
Value: 0 = Disabled. 1 = Enabled. Default is 0.
Optional: Yes

port-member-list

Description: A list of logical switch port interface (fibre channel and flex) names.
Flag: read-write
This container has the following leaf:

port-member

Description: The name of the fibre channel interface including the slot and port number (slot/port). This list only contains fibre channel ports and flex ports.
Flag: read-write
Type: fibrechannel:slot-port-name-type
Value: The fibre channel or flex port name (slot/port_name).
Optional: Yes

ge-port-member-list

Description: A list of logical switch port interface Gigabit Ethernet (GE) names.
Flag: read-write
This container has the following leaf:

port-member

Description: The name of the Gigabit Ethernet (GE) interface including the slot and port number (slot/port). This list only contains GE ports.
Flag: read-write
Type: fibrechannel:slot-port-name-type
Value: The GE port name (slot/port_name).
Optional: Yes

Supported methods

The GET, POST, PATCH, DELETE, OPTIONS, and HEAD operations are supported in this module.

Examples

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>10:00:00:05:1e:b7:44:00</switch-wwn>
        <default-switch-status>1</default-switch-status>
        <logical-isl-enabled>1</logical-isl-enabled>
        <ficon-mode-enabled>0</ficon-mode-enabled>
        <port-member-list>
            <port-member>7/0</port-member>
```

```
            <port-member>7/1</port-member>
            <port-member>7/2</port-member>
            <port-member>7/3</port-member>
            <port-member>7/4</port-member>
            <port-member>7/5</port-member>
            <port-member>7/6</port-member>
            <port-member>7/7</port-member>
            <port-member>7/8</port-member>
            <port-member>7/9</port-member>
            <port-member>7/10</port-member>
        </port-member-list>
        <ge-port-member-list/>
    </fibrechannel-logical-switch>
    <fibrechannel-logical-switch>
        <fabric-id>10</fabric-id>
        <base-switch-enabled>0</base-switch-enabled>
        <switch-wwn>10:00:00:05:1e:b7:44:01</switch-wwn>
        <default-switch-status>0</default-switch-status>
        <logical-isl-enabled>1</logical-isl-enabled>
        <ficon-mode-enabled>0</ficon-mode-enabled>
        <port-member-list>
            <port-member>3/12</port-member>
            <port-member>3/13</port-member>
            <port-member>3/14</port-member>
            <port-member>3/15</port-member>
        </port-member-list>
        <ge-port-member-list/>
    </fibrechannel-logical-switch>
    <fibrechannel-logical-switch>
        <fabric-id>20</fabric-id>
        <base-switch-enabled>0</base-switch-enabled>
        <switch-wwn>10:00:00:05:1e:b7:44:02</switch-wwn>
        <default-switch-status>0</default-switch-status>
        <logical-isl-enabled>1</logical-isl-enabled>
        <ficon-mode-enabled>0</ficon-mode-enabled>
        <port-member-list/>
        <ge-port-member-list/>
    </fibrechannel-logical-switch>
    <fibrechannel-logical-switch>
        <fabric-id>1</fabric-id>
        <base-switch-enabled>1</base-switch-enabled>
        <switch-wwn>10:00:00:05:1e:b7:44:03</switch-wwn>
        <default-switch-status>0</default-switch-status>
        <logical-isl-enabled>0</logical-isl-enabled>
        <ficon-mode-enabled>0</ficon-mode-enabled>
        <port-member-list/>
        <ge-port-member-list/>
    </fibrechannel-logical-switch>
    <fibrechannel-logical-switch>
        <fabric-id>2</fabric-id>
        <base-switch-enabled>0</base-switch-enabled>
        <switch-wwn>10:00:00:05:1e:b7:44:04</switch-wwn>
        <default-switch-status>0</default-switch-status>
```

```
            <logical-isl-enabled>1</logical-isl-enabled>
            <ficon-mode-enabled>0</ficon-mode-enabled>
            <port-member-list/>
            <ge-port-member-list/>
        </fibrechannel-logical-switch>
        <fibrechannel-logical-switch>
            <fabric-id>3</fabric-id>
            <base-switch-enabled>0</base-switch-enabled>
            <switch-wwn>10:00:00:05:1e:b7:44:05</switch-wwn>
            <default-switch-status>0</default-switch-status>
            <logical-isl-enabled>0</logical-isl-enabled>
            <ficon-mode-enabled>1</ficon-mode-enabled>
            <port-member-list/>
            <ge-port-member-list/>
        </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
```

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>10</fabric-id>
        <base-switch-enabled>0</base-switch-enabled>
        <switch-wwn>10:00:00:05:1e:b7:44:01</switch-wwn>
        <default-switch-status>0</default-switch-status>
        <logical-isl-enabled>1</logical-isl-enabled>
        <ficon-mode-enabled>0</ficon-mode-enabled>
        <port-member-list>
            <port-member>3/12</port-member>
            <port-member>3/13</port-member>
            <port-member>3/14</port-member>
            <port-member>3/15</port-member>
        </port-member-list>
        <ge-port-member-list/>
    </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 3 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 or 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.

> **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.

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 2 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 the 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 the Logical ISL on a Logical Switch

Logical ISLs can be disabled during logical switch creation with the POST request, and later 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 Logical ISL on a Logical Switch

You can only disable logical ISL during logical switch creation. You must use a POST request to disable logical ISL and create a new logical switch.

> **NOTE**
> You cannot disable logical ISL on an existing logical switch.

This example uses a POST request to disable 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.

History

Release version	History
Fabric OS 8.2.0a	This API call was introduced.
Fabric OS 8.2.1b	The top-level container name changed from "logical-switch" to "brocade-fibrechannel-logical-switch". The previous top-level container name "logical-switch" is still supported in this release.

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.

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 Types for 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}?
          x--rw enabled-state?                uint32
          +--rw is-enabled-state?             boolean
          +--ro operational-status?           uint32
          +--ro up-time?                      yang:timeticks
          +--ro model?                        string
          +--ro firmware-version?             string
          +--rw ip-address
          |  +--rw ip-address*                        inet:ip-address
          +--rw ip-static-gateway-list
          |  +--rw ip-static-gateway*         fru:ip-gateway-type
          +--rw subnet-mask?                  inet:ipv4-address
          +--rw domain-name?                  string
          +--rw dns-servers
          |  +--rw dns-server*                inet:ip-address
          +--rw fabric-user-friendly-name?    string {fibrechannel:fibrechannel_switch_platform}?
          +--rw ag-mode?                      uint8
          +--rw banner?                       string
```

URI Format

The URI format for this module takes the following form:

<base_URI>`/running/brocade-fibrechannel-switch` followed by the leafs as listed in the module tree.

Parameters

NOTE
The top-level container name changed from "switch" to "brocade-fibrechannel-switch". The previous top-level container name "switch" is still supported in this release.

brocade-fibrechannel-switch
Description: Switch state parameters.
Flag: read-write
Optional: No
This container has the following leafs:

fibrechannel-switch
Description: Switch identification parameters.
Requests can also be made using a query that specifies the Virtual Fabric ID of the fabric. This request also provides the switch state parameters.
Flag: read-write
Optional: No
Key: *name*
This list has the following leafs:

name
Description: The switch world wide name (WWN).
Flag: read-write
Type: fibrechannel:wwn-type
Values: A valid WWN name.
Optional: No

domain-id
Description: Domain ID of the switch.
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.
Flag: read-write
Type: fibrechannel:domain-id-type
Values: 1 through 239.
Optional: Yes

user-friendly-name
Description: The ASCII name assigned to the switch by the administrator.
Flag: read-write
Type: string
Value: 1 to 30 alphanumeric characters plus hyphens (-), periods (.), and underscores (_). Spaces are not allowed. A switch name can begin with either a letter or number, but a switch name that begins with a numeric (0-9) character must also have at least an underscore (_), hyphen (-), period (.), or alphabetic (A-Z, a-z) character. A switch name with only numeric characters is not valid.
Optional: Yes

fcid
Description: This parameter is deprecated. Use the fcid-hex parameter. The destination ID (D_ID) of the switch (decimal format).
Flag: read-only
Type: fibrechannel:fcid-type
Config: false
Value: A valid destination FCID value.
Optional: Yes

fcid-hex

Description: The destination ID (D_ID) of the switch (hexidecimal format).
Flag: read-only
Type: fibrechannel:fcid-hex-string-type
Config: false
Value: A valid D_ID of the port.
Optional: Yes

vf-id

Description: The Virtual Fabric identifier (VFID) of the logical switch.
Flag: read-only
Type: int16
Config: false
Value: A valid VFID value from 1 to 128 or -1 = Indicates that the Virtual Fabric feature is unsupported or disabled.
Optional: Yes

principal

Description: Indicates the principal switch in the fabric.
Flag: read-only
Type: uint8
Config: false
Value: 0 = This is not the principal switch in the fabric. 1 = This is the principal switch in the fabric.
Optional: Yes

enabled-state

Description: This parameter is deprecated. Use the *is-enabled-state* leaf to configure and the *operational-status* leaf to obtain the current state of the switch. The current state of the switch.
Flag: read-write
Type: uint32
Values: 0 = Undefined. 2 = Switch is enabled. 3 = Switch is disabled. 7 = Switch is being tested. This leaf can be set to 2 or 3 only; it cannot be set to 0 or 7.
Default: 2
Optional: Yes

is-enabled-state

Description: The current state of the switch.
Flag: read-write
Type: boolean
Values: true = Switch is enabled. false = Switch is disabled.
Optional: Yes

operational-status

Description: The current state of the switch.
Flag: read-write
Type: uint32
Config: false
Values: 0 = Undefined. 7 = Switch is being tested.
This leaf can be set to 2 or 3 only; it cannot be set to 0 or 7.
Default: 2
Optional: Yes

up-time

Description: The period of time elapsed since the last reboot of the specified switch.

The absolute time interval for which the switch has been up, not the time measured between midnight, January 1, 1970 UTC. and the most recent reboot of the switch.
Flag: read-only
Type: yang:timeticks
Config: false
Units: seconds
Values: 0 through 4294967296.
Optional: Yes

model
Description: The model name of the switch.
Flag: read-only
Type: string
Config: false
Value: The Brocade switch type of the switch (example: 145).
Optional: Yes

firmware-version
Description: A human-readable string identifying the firmware version running on the switch.
Flag: read-only
Type: string
Config: false
Value: The Brocade-defined firmware version ID.
Optional: Yes

ip-address
Description: A list of out-of-band IP addresses assigned to the Ethernet port on the switch. This container is needed by the underlying infrastructure.
Flag: read-write
This container has the following leaf:

ip-address
Description: A list of IP addresses assigned to the Ethernet port on the switch.
Flag: read-write
Type: inet:ip-address
Value: A valid IPv4 or IPv6 address.
Optional: No

ip-static-gateway-list
Description: A list of static gateway IPv4 and IPv6 addresses for the switch IP address.
Flag: read-write
This container has the following leaf:

ip-static-gateway
Description: 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.
Flag: read-write
Type: fru:ip-gateway-type
Value: A valid static gateway IPv4 or IPv6 address.
Optional: Yes

subnet-mask
Description: The IPv4 subnet mask of the switch IP network.
Flag: read-write
Type: inet:ipv4-address

Value: A valid IPv4 address.
Optional: Yes

domain-name
Description: The DNS domain name of the switch. Note that leaving this field empty clears the DNS domain name.
Flag: read-write
Type: string
Value: 4 to 63 alphanumeric characters. The DNS domain name must start with a alphabetic character (a-z, A-Z) and must not contain blank spaces.
Optional: Yes

dns-servers
Description: A list of DNS server addresses.
Flag: read-write
This container has the following leaf:

dns-server
Description: A list of DNS server addresses which can handle the mapping of the domain names to the IP address of an internet resource which is needed by various networking protocols. Brocade FC switches support a maximum of two DNS servers.
Flag: read-write
Type: inet:ip-address
Value: A valid IP address.
Optional: Yes

fabric-user-friendly-name
Description: The human-readable name for the fabric.
Flag: read-write
Type: string
Value: 1 to 128 alphanumeric characters, plus hyphens (-), periods (.), and underscores (_). The value is not case-sensitive.
Optional: Yes

ag-mode
Description: Indicates the Access Gateway (AG) mode capability and enablement state. A switch is capable of AG mode support when this value is not zero.
Flag: read-write
Type: uint8
Value: 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.
Optional: Yes

banner
Description: The text that displays during the log on process.
Flag: read-write
Type: string
Value: 1 to 1022 alphanumeric characters. If the banner text exceeds the maximum limit, the software truncates the banner text.
Optional: Yes

Supported Methods

Only the GET, OPTIONS, PATCH, and HEAD method operations are supported in this module.

Examples

Retrieving Switch Resource Information

This example uses a GET request to retrieve the switch resources.

Structure

GET *<base_URI>*/running/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.

```
<?xml version="1.0"?>
<Response>
    <fibrechannel-switch>
        <name>10:10:10:f5:7c:4a:ac:6c</name>
        <domain-id>1</domain-id>
        <fcid>16776193</fcid>
        <fcid-hex>0xfffc01</fcid-hex>
        <user-friendly-name>X6-4_066_243</user-friendly-name>
        <enabled-state>2</enabled-state>
        <banner>Banner test message One</banner>
        <up-time>7240</up-time>
        <domain-name/>
        <dns-servers/>
        <principal>0</principal>
        <ip-address>
            <ip-address>10.20.10.243</ip-address>
        </ip-address>
        <subnet-mask>255.255.255.0</subnet-mask>
        <model>165.0</model>
        <firmware-version>v8.2.1_bld42</firmware-version>
        <ip-static-gateway-list>
            <ip-static-gateway>10.38.64.1</ip-static-gateway>
        </ip-static-gateway-list>
        <vf-id>128</vf-id>
        <fabric-user-friendly-name>East</fabric-user-friendly-name>
        <ag-mode>0</ag-mode>
    </fibrechannel-switch>
</Response>
```

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 an VFID of 20.

Structure

GET *<base_URI>*/running/fibrechannel-switch?vf-id=*<vf-id#>*

URI

```
GET https://10.10.10.10/rest/running/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>10:10:10:f5:7c:00:c5:30</name>
        <domain-id>232</domain-id>
        <fcid>16776424</fcid>
        <fcid-hex>0xfffce8</fcid-hex>
        <user-friendly-name>G630_066_232</user-friendly-name>
        <enabled-state>2</enabled-state>
        <banner/>
        <up-time>91593</up-time>
        <domain-name>brm.bsnlab.broadcom.net</domain-name>
        <dns-servers>
            <dns-server>10.10.10.10</dns-server>
            <dns-server>10.10.10.30</dns-server>
        </dns-servers>
        <principal>0</principal>
        <ip-address>
            <ip-address>10.10.10.232</ip-address>
        </ip-address>
        <subnet-mask>255.255.255.0</subnet-mask>
        <model>173.0</model>
        <firmware-version>v8.2.1_bld42</firmware-version>
        <ip-static-gateway-list>
            <ip-static-gateway>10.10.10.1</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>
    </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.

History

Release Version	History
Fabric OS 8.2.0	This API call was introduced.
Fabric OS 8.2.1	Added enabled and disable support for ag-mode. Added ip-static-gateway-list, subnet-mask, and banner leafs.
Fabric OS 8.2.1b	The top-level container name changed from "switch" to "brocade-fibrechannel-switch". The previous top-level container name "switch" is still supported in this release.

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.

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 Types

```
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-throughput?           uint64
     |  +--ro tx-percentage?           fibrechannel:percentage-type
     |  +--ro rx-bandwidth?            uint32
     |  +--ro rx-throughput?           uint64
     |  +--ro rx-percentage?           fibrechannel:percentage-type
     |  +--ro txrx-bandwidth?          uint32
     |  +--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
```

for data type descriptions.

URI Format

The URI format for this module takes one of the following forms:

- `<base_URI>/running/brocade-interface/fibrechannel/name/fibre_channel_interface_name/trunk-port-enabled/` to view the trunk settings for a physical port or to enable or disable trunking on a port.
- `<base_URI>/running/brocade-fibrechannel-trunk/trunk` followed by the leafs as listed in the module tree to view trunk data for all trunks on a device.

• *<base_URI>*/running/brocade-fibrechannel-trunk/performance followed by the leafs as listed in the module tree to view performance data for trunks on a device.

• *<base_URI>*/running/brocade-fibrechannel-trunk/trunk-area followed by the leafs as listed in the module tree to view data for all port trunk area groups on a switch or to create, modify, or delete a trunk area.

Supported Methods

Only the OPTIONS, GET, DELETE, HEAD, and POST operations are supported in this module.

History

Release Version	History
Fabric OS 8.2.1	This API call was introduced.
Fabric OS 9.0.0	This API call was modified to add the trunk-type parameter to the trunk list.
Fabric OS 9.1.0	This API call was modified to edit the master, neighbor-switch-name, and master-port leafs.

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.

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 Types for data type descriptions.

```
module: brocade-fru
    +--rw brocade-fru
       +--rw blade* [slot-number]
       |  +--rw slot-number                  uint16
       |  +--ro manufacturer?                string
       |  +--ro blade-type?                  string
       |  +--ro blade-id?                    uint16
       |  +--ro blade-state?                 string
       |  +--ro model-name?                  string
       |  +--ro firmware-version?            fru:firmware-version-type
       |  +--ro fc-port-count?               uint16
       |  +--ro ge-port-count?               uint16
       |  +--ro ip-address-list
       |  |  +--ro ip-address*   inet:ip-address
       |  +--ro ip-gateway-list
       |  |  +--ro ip-gateway*   fru:ip-gateway-type
       |  +--ro subnet-mask?                 inet:ip-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 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 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 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 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
```

URI Format

The URI format for this module takes the following form:

• *<base_URI>*/running/brocade-fru/blade followed by the leafs as listed in the module tree to view information about all blades in the chassis.

• *<base_URI>*/running/brocade-fru/blade/slot-number/*slot-number* to view information about the specified blade.

• *<base_URI>*/running/brocade-fru/fan followed by the leafs as listed in the module tree to view information about all fans in the chassis.

• *<base_URI>*/running/brocade-fru/fan/unit-number/*fan-unit* to view information about the specified fan unit.

• *<base_URI>*/running/brocade-fru/power-supply followed by the leafs as listed in the module tree to view information about all power supplies in the chassis.

• *<base_URI>*/running/brocade-fru/power-supply/unit-number/*power-supply-unit* to view information about the specified power-supply unit.

Parameters

brocade-fru

Description: Configuration and runtime information of the FRUs installed in the chassis.
Flag: read-write
This container has the following leafs:

blade

Description: A list of blade details for the specified slot number.
Flag: read-write
Key: *slot-number*
This list has the following leafs:

slot-number

Description: The number of the physical slot in the chassis in which the blade is inserted.
Flag: read-write
Type: uint16
Value: 0 through 12. 0 for a fixed-port switch. 1 through 12 for a chassis
Optional: Yes

manufacturer

Description: The name of the blade manufacturer.
Flag: read-only

Type: string
Config: false
Value: 1 to 63 alphanumeric characters.
Optional: Yes

blade-type
Description: The type of blade (sw blade, cp blade, core blade).
Flag: read-only
Type: string
Config: false
Value: 1 to 32 alphanumeric characters.
Optional: Yes

blade-id
Description: The ID of the blade. This parameter is available only when the blade type is not unknown (blade-type != 'unknown').
Flag: read-only
Type: uint16
Config: false
Value: 1 to 1000.
Optional: Yes

blade-state
Description: The current state of the blade.
Flag: read-only
Type: string
Config: false
Value: 1 to 255 characters.
Optional: Yes

model-name
Description: A printable ASCII string that describes the model of the blade. This parameter is available only when the blade type is not unknown (blade-type != 'unknown') or the slot number is not zero (slot-number != 0).
Flag: read-only
Type: string
Config: false
Value: 1 to 255 printable ASCII characters.
Optional: Yes

firmware-version
Description: 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.
Flag: read-only
Type: fru:firmware-version-type
Config: false
Value: 1 to 255 printable ASCII characters.
Optional: Yes

fc-port-count
Description: The number of FC ports supported by the blade. This parameter is available only when the blade type is not CP or not unknown (blade-type != 'cp blade' or blade-type != 'unknown').
Flag: read-only
Type: uint16

Config: false
Value: 0 to 256.
Optional: Yes

ge-port-count
Description: The number of GE ports supported by the blade. This parameter is available only when extension is enabled or the blade ID is 75 (extension-enabled = true or blade-id = 75).
Flag: read-only
Type: uint16
Config: false
Value: 0 to 256.
Optional: Yes

ip-address-list
Description: A list of IP addresses assigned to the CP blade. This parameter is available only for CP blades (blade-type = 'cp blade').
Flag: read-only
Config: false
This list has the following leaf:

ip-address
Description: The list of IP addresses assigned to the CP blade.
Flag: read-only
Type: inet:ip-address
Value: A valid IPv4 or IPv6 address.
Optional: Yes

ip-gateway-list
Description: A list of gateway IP addresses of the CP blade. This parameter is available only for CP blades (blade-type = 'cp blade').
Flag: read-only
Config: false
This list has the following leaf:

ip-gateway
Description: 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 available only for CP blades (blade-type = 'cp blade').
Flag: read-only
Type: fru:ip-gateway-type
Value: A valid IPv4 or IPv6 address.
Optional: Yes

subnet-mask
Description: The subnet mask of the network. This parameter is available only for CP blades (blade-type = 'cp blade').
Flag: read-only
Type: inet:ip-address
Value: A valid IPv4 address.
Optional: Yes

primary-firmware-version
Description: A printable ASCII string that identifies the firmware version running on the primary partition of the CP blade. This parameter is available only for CP blades (blade-type = 'cp blade').
Flag: read-only
Type: fru:firmware-version-type

Config: false
Value: 1 to 255 printable ASCII characters.
Optional: Yes

power-consumption
Description: The maximum power consumption allocated for the blade.
Flag: read-only
Type: fru:power-detail-type
Config: false
Value: The maximum power consumption allocated for the blade.
Optional: Yes

power-usage
Description: The real-time power consumed by the FRU.
Flag: read-only
Type: fru:power-detail-type
Config: false
Value: The real-time power consumed by the FRU.
Optional: Yes

secondary-firmware-version
Description: A printable ASCII string that identifies the firmware version running on the secondary partition of the CP blade. This parameter is available only for CP blades (blade-type = 'cp blade').
Flag: read-only
Type: fru:firmware-version-type
Config: false
Value: 1 to 255 printable ASCII characters.
Optional: Yes

extension-enabled
Description: Whether the switch or blade supports extension. For blade IDs 154, 186, and 213, the value is set to enabled (true).
Flag: read-only
Type: boolean
Config: false
Value: true = Extension is supported. **false** = Extension is not suppported.
Optional: Yes

extension-app-mode
Description: The application mode configuration of the extension blade or switch. This parameter is available only when extension is enabled (extension-enabled = true). Note that a change in the application mode configuration is a disruptive operation and that when the configuration change is successful, a switch automatically reboots and a blade automatically powers off and on. Blade ID 213 supports only hybrid mode. For blade ID 213, the default is hybrid mode. For blade ID 154 and 186, the default is FCIP mode.
Flag: read-write
Type: string
Value: FCIP = Supports FCIP-only tunnels. **hybrid** = Supports FCIP with IP Extension tunnels. **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 a slot power off or a slot faulty after reboot or powercycle.
Optional: Yes

extension-ve-mode

Description: The VE mode configuration of the extension blade or switch. This parameter is available only when extension is enabled (extension-enabled = true). Note that a VE mode configuration change is a disruptive operation and that when the configuration change is successful, a switch automatically reboots and 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.
Flag: read-write
Type: string
Value: not applicable = Not a supported configuration. **10VE** = 10 VEs available for tunnel configuration (5 VE_Ports per DP). 10VE mode is required for hybrid mode operation. **20VE** = 20 VEs available for tunnel configuration (10 VE_Ports per DP). 20VE mode is allowed in FCIP mode only. **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 a slot power off or a slot faulty after reboot or powercycle.
Optional: Yes

extension-ge-mode

Description: The GE mode configuration of the extension blade or switch. This parameter is applicable only on the Brocade 7810 Extension 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.
Flag: read-write
Type: string
Value: not applicable = Not a supported configuration. **optical** = ge2 and ge3 use the first two optical ports. **copper** = ge0 and ge1 use the front-end RJ-45 ports. The first two optical ports are disabled. **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 a slot power off or a slot faulty after reboot or powercycle.
Optional: Yes

time-alive

Description: The number of days the FRU has been powered on.
Flag: read-only
Type: fru:time-detail-type
Config: false
Value: The number of days the FRU has been powered on.
Optional: Yes

time-awake

Description: The number of days since the FRU was last powered on.
Flag: read-only
Type: fru:time-detail-type
Config: false
Value: The number of days since the FRU was last powered on.
Optional: Yes

part-number

Description: The part number assigned by the organization responsible for producing or manufacturing of the FRU.
Flag: read-only
Type: fru:part-number-type
Config: false
Value: The part number assigned by the organization responsible for producing or manufacturing of the FRU.

Optional: Yes

serial-number

Description: The serial-number resource contains a Printable ASCII String that specifies the serial number of the FRU.
Flag: read-only
Type: fru:serial-number-type
Config: false
Value: The serial-number resource contains a Printable ASCII String that specifies the serial number of the FRU.
Optional: Yes

fan

Description: The details about the fan units.
Flag: read-only
Key: unit-number
Config: false
This list has the following leafs:

unit-number

Description: The physical slot number in the chassis where the fan is located.
Flag: read-only
Type: uint16
Value: 1 to 16.
Optional: Yes

power-consumption

Description: The maximum power consumption allocated for the fan in watts.
Flag: read-only
Type: fru:power-detail-type
Value: —1000 to —1 watts.
Optional: Yes

operational-state

Description: The current operational state of the fan.
Flag: read-only
Type: string
Value: The operational state of the fan (faulty, ok, below minimum, above maximum, absent, unknown, or not ok)
Optional: Yes

speed

Description: The fan speed in RPM.
Flag: read-only
Type: uint32
Value: 1 to 100000 RPM.
Optional: Yes

airflow-direction

Description: The air flow direction of the fan blowers.
Flag: read-only
Type: air-flow-direction-type
Value: The air flow direction of the fan blowers (forward (non-portside exhaust) ,non-portside exhaust, reverse (non-portside intake), non-portside intake, and not available).
Optional: Yes

time-alive
Description: The number of days the FRU has been powered on.
Flag: read-only
Type: fru:time-detail-type
Config: false
Value: The number of days the FRU has been powered on.
Optional: Yes

time-awake
Description: The number of days since the FRU was last powered on.
Flag: read-only
Type: fru:time-detail-type
Config: false
Value: The number of days since the FRU was last powered on.
Optional: Yes

part-number
Description: The part number assigned by the organization responsible for producing or manufacturing of the FRU.
Flag: read-only
Type: fru:part-number-type
Config: false
Value: The part number assigned by the organization responsible for producing or manufacturing of the FRU.
Optional: Yes

serial-number
Description: The serial-number resource contains a Printable ASCII String that specifies the serial number of the FRU.
Flag: read-only
Type: fru:serial-number-type
Config: false
Value: The serial-number resource contains a Printable ASCII String that specifies the serial number of the FRU.
Optional: Yes

power-supply
Description: The details about the power supply units.
Flag: read-only
Key: unit-number
This list has the following leafs:

unit-number
Description: The physical slot number of the chassis where the power supply is located.
Flag: read-only
Type: uint16
Value: 1 to 16.
Optional: Yes

power-production
Description: The maximum power production allocated for the power supply unit in watts.
Flag: read-only
Type: uint32
Value: 0 to 10000 watts.
Optional: Yes

input-voltage
Description: The input voltage of the power supply unit in volts.
Flag: read-only
Type: decimal64
Value: 0 to 900.99 volts.
Optional: Yes

airflow-direction
Description: The air flow direction of the power supply fans.
Flag: read-only
Type: air-flow-direction-type
Value: The air flow direction of the fan blowers (forward (non-portside exhaust) ,non-portside exhaust, reverse (non-portside intake), non-portside intake, and not available).
Optional: Yes

power-source
Description: The power supply input voltage type.
Flag: read-only
Type: string
Value: The input voltage type (such as AC, DC, HVAC, HVDC, or not available).
Optional: Yes

operational-state
Description: The operational state of the power supply.
Flag: read-only
Type: string
Value: The operational state of the power supply (such as absent, ok, faulty, predicting failure, unknown, or try reseating unit).
Optional: Yes

temperature
Description: The temperature of the power supply sensor in centigrade.
Flag: read-only
Type: decimal64
Value: 0 to 300.00 centigrade.
Optional: Yes

power-usage
Description: The real-time power consumed by the FRU.
Flag: read-only
Type: fru:power-detail-type
Config: false
Value: The real-time power consumed by the FRU.
Optional: Yes

time-alive
Description: The number of days the FRU has been powered on.
Flag: read-only
Type: fru:time-detail-type
Config: false
Value: The number of days the FRU has been powered on.
Optional: Yes

time-awake
Description: The number of days since the FRU was last powered on.
Flag: read-only
Type: fru:time-detail-type

Config: false
Value: The number of days since the FRU was last powered on.
Optional: Yes

part-number
Description: The part number assigned by the organization responsible for producing or manufacturing of the FRU.
Flag: read-only
Type: fru:part-number-type
Config: false
Value: The part number assigned by the organization responsible for producing or manufacturing of the FRU.
Optional: Yes

serial-number
Description: The serial-number resource contains a Printable ASCII String that specifies the serial number of the FRU.
Flag: read-only
Type: fru:serial-number-type
Config: false
Value: The serial-number resource contains a Printable ASCII String that specifies the serial number of the FRU.
Optional: Yes

Supported Methods

Only the GET, PATCH, and HEAD operations are supported in this module.

Examples

Viewing Data for all Blades in a Chassis

The following example uses the GET request to view data on all blades in the chassis.

Structure

GET *<base_URI>*/running/brocade-fru/blade

URI

```
GET https://10.10.10.10/rest/running/brocade-fru/blade
```

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>
    <blade>
        <slot-number>1</slot-number>
        <part-number>60-1003201-12</part-number>
        <serial-number>DYK3213M01R</serial-number>
        <manufacturer>Brocade Communications Systems LLC</manufacturer>
        <blade-type>cp blade</blade-type>
        <firmware-version>v8.2.1b_rc1_bld12</firmware-version>
```

```
            <primary-firmware-version>v8.2.1b_rc1_bld12</primary-firmware-version>
            <secondary-firmware-version>v8.2.1b_rc1_bld12</secondary-firmware-version>
            <ip-address-list>
                <ip-address>10.38.66.241</ip-address>
                <ip-address>2620:100:4:fa01:c6f5:7cff:fe44:a5d9/64</ip-address>
            </ip-address-list>
            <subnet-mask>255.255.240.0</subnet-mask>
            <ip-gateway-list>
                <ip-gateway>10.38.64.1</ip-gateway>
                <ip-gateway>fe80::126</ip-gateway>
            </ip-gateway-list>
            <model-name>CPX6</model-name>
            <extension-enabled>false</extension-enabled>
            <blade-id>175</blade-id>
            <blade-state>enabled</blade-state>
            <power-usage>-48</power-usage>
            <time-alive>435</time-alive>
            <time-awake>2</time-awake>
            <power-consumption>-50</power-consumption>
        </blade>
        <blade>
            <slot-number>2</slot-number>
            <part-number>60-1003201-12</part-number>
            <serial-number>DYK3213M03G</serial-number>
            <manufacturer>Brocade Communications Systems LLC</manufacturer>
            <blade-type>cp blade</blade-type>
            <firmware-version>v8.2.1b_rc1_bld12</firmware-version>
            <primary-firmware-version>v8.2.1b_rc1_bld12</primary-firmware-version>
            <secondary-firmware-version>v8.2.1b_rc1_bld12</secondary-firmware-version>
            <ip-address-list>
                <ip-address>10.38.66.242</ip-address>
                <ip-address>2620:100:4:fa01:c6f5:7cff:fe44:ac91/64</ip-address>
            </ip-address-list>
            <subnet-mask>255.255.240.0</subnet-mask>
            <ip-gateway-list>
                <ip-gateway>10.38.64.1</ip-gateway>
                <ip-gateway>fe80::126</ip-gateway>
            </ip-gateway-list>
            <model-name>CPX6</model-name>
            <extension-enabled>false</extension-enabled>
            <blade-id>175</blade-id>
            <blade-state>enabled</blade-state>
            <power-usage>-48</power-usage>
            <time-alive>435</time-alive>
            <time-awake>2</time-awake>
            <power-consumption>-50</power-consumption>
        </blade>
    .
    .
    .
        <blade>
            <slot-number>8</slot-number>
            <part-number>60-1003847-01</part-number>
```

```
        <serial-number>FDU0325N021</serial-number>
        <manufacturer>Brocade Communications Systems LLC</manufacturer>
        <blade-type>sw blade</blade-type>
        <firmware-version>v8.2.1b_rc1_bld12</firmware-version>
        <model-name>FC32-64</model-name>
        <fc-port-count>64</fc-port-count>
        <extension-enabled>false</extension-enabled>
        <blade-id>204</blade-id>
        <blade-state>enabled</blade-state>
        <power-usage>-181</power-usage>
        <time-alive>379</time-alive>
        <time-awake>2</time-awake>
        <power-consumption>-387</power-consumption>
    </blade>
</Response>
```

Viewing Data for a Specific Blade

The following example uses the GET request to view data for a specific blade (6).

Structure

GET *<base_URI>*/running/ brocade-fru/blade/slot-number/*slot-number*

URI

```
GET https://10.10.10.10/rest/running/brocade-fru/blade/slot-number/6
```

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>
    <blade>
        <slot-number>5</slot-number>
        <part-number>60-1003226-10</part-number>
        <serial-number>DZD3213M011</serial-number>
        <manufacturer>Brocade Communications Systems LLC</manufacturer>
        <blade-type>core blade</blade-type>
        <firmware-version>v8.2.1b_rc1_bld12</firmware-version>
        <model-name>CR32-4</model-name>
        <fc-port-count>32</fc-port-count>
        <extension-enabled>false</extension-enabled>
        <blade-id>176</blade-id>
        <blade-state>enabled</blade-state>
        <power-usage>-106</power-usage>
        <time-alive>435</time-alive>
        <time-awake>2</time-awake>
        <power-consumption>-244</power-consumption>
    </blade>
</Response>
```

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 all Power Supplies in a Chassis

The following example uses the GET request to view data on all power supplies in the chassis.

Structure

GET *<base_URI>*/running/brocade-fru/power-supply

URI

```
GET https://10.10.10.10/rest/running/brocade-fru/power-supply
```

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>211.50</input-voltage>
        <temperature>28.00</temperature>
        <time-alive>413</time-alive>
        <time-awake>2</time-awake>
    </power-supply>
    <power-supply>
        <unit-number>2</unit-number>
        <serial-number>DUC2M32M2ME</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>212.00</input-voltage>
        <temperature>29.00</temperature>
        <time-alive>432</time-alive>
        <time-awake>2</time-awake>
    </power-supply>
</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>211.50</input-voltage>
        <temperature>28.00</temperature>
        <time-alive>413</time-alive>
        <time-awake>2</time-awake>
    </power-supply>
</Response>
```

History

Release Version	History
Fabric OS 8.2.1	This API call was introduced.
Fabric OS 8.2.1b	Added the power-consumption, power-usage, time-alive, and time-awake parameters.

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, then 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.

> **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-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                              string
       |  +--ro wwn?                              fibrechannel:wwn-type
       |  +--ro operational-status?               uint32
       |  x--rw enabled-state?                    uint32
       |  +--rw is-enabled-state?                 boolean
       |  +--rw user-friendly-name?               string
       |  +--rw speed?                            brocade-interface-types:fc-speed-type
       |  +--ro max-speed?                        brocade-interface-types:fc-speed-type
       |  +--ro auto-negotiate?                   uint8
       |  +--rw g-port-locked?                    uint8 {fibrechannel:fibrechannel_switch_platform}?
       |  +--rw e-port-disable?                   uint8 {fibrechannel:fibrechannel_switch_platform}?
       |  +--rw n-port-enabled?                   uint8 {fibrechannel:access_gateway_platform}?
       |  +--rw d-port-enable?                    uint8
       |  +--rw persistent-disable?               uint8
       |  +--ro neighbor
       |  |  +--ro wwn*   fibrechannel:wwn-type
       |  +--ro neighbor-node-wwn?                fibrechannel:wwn-type
       |  x--ro fcid?                             fibrechannel:fcid-type
       |  +--ro fcid-hex?                         fibrechannel:fcid-hex-string-type
       |  +--ro port-type?                        brocade-interface-types:port-type-type
       |  +--rw qos-enabled?                      uint8
       |  +--rw compression-configured?           uint8
       |  +--ro compression-active?               uint8
       |  +--ro encryption-active?                uint8 {fibrechannel:fibrechannel_switch_platform}?
       |  +--rw target-driven-zoning-enable?      uint8
       |  +--rw sim-port-enabled?                 uint8 {fibrechannel:fibrechannel_switch_platform}?
       |  +--rw mirror-port-enabled?              int8
       |  +--rw credit-recovery-enabled?          int8
       |  +--ro credit-recovery-active?           int8
       |  +--ro fec-active?                       int8
```

```
|  +--rw f-port-buffers?                  uint8
|  +--rw e-port-credit?                   uint8
|  +--rw csctl-mode-enabled?              int8 {fibrechannel:fibrechannel_switch_platform}?
|  +--rw fault-delay-enabled?             uint8
|  +--rw octet-speed-combo?               int8
|  +--ro long-distance?                   uint8 {fibrechannel:fibrechannel_switch_platform}?
|  +--ro vc-link-init?                    uint8 {fibrechannel:fibrechannel_switch_platform}?
|  +--rw isl-ready-mode-enabled?          uint8 {fibrechannel:fibrechannel_switch_platform}?
|  +--rw rscn-suppression-enabled?        uint8 {fibrechannel:fibrechannel_switch_platform}?
|  +--rw los-tov-mode-enabled?            uint8
|  +--rw npiv-enabled?                    uint8
|  +--rw npiv-pp-limit?                   int16
|  +--ro npiv-flogi-logout-enabled?       int8
|  +--rw ex-port-enabled?                 uint8 {fibrechannel:fibrechannel_switch_platform}?
|  +--rw fec-enabled?                     int8
|  +--ro via-tts-fec-enabled?             int8
|  +--rw port-autodisable-enabled?        uint8
|  +--rw rate-limit-enabled?              int16 {fibrechannel:fibrechannel_switch_platform}?
|  +--rw non-dfe-enabled?                 uint8 {fibrechannel:fibrechannel_switch_platform}?
|  +--rw trunk-port-enabled?              uint8
|  +--ro default-index?                   fibrechannel:user-port-number-type
|  +--ro physical-state?                  string
|  +--ro pod-license-status?              boolean
+--rw fibrechannel-statistics* [name]
|  +--rw name                                        string
|  +--ro sampling-interval?                          uint16
|  +--ro time-generated?                             fibrechannel:time-generated-type
|  +--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
+--rw extension-ip-interface* [name ip-address dp-id]
...
+--rw gigabitethernet* [name]
...
+--rw gigabitethernet-statistics* [name]
   ...
+--ro logical-e-port* [port-index] {fibrechannel:fibrechannel_extended_isl}?
   |  +--ro port-index                  fibrechannel:user-port-number-type
   |  +--ro fabric-id?                  fibrechannel:fabric-id-type
   |  +--ro operational-status?         enumeration
   |  +--ro offline-reason?             enumeration
   |  +--ro neighbor-node-wwn?          fibrechannel:wwn-type
   |  +--ro associated-physical-ports
   |     +--ro port*   fibrechannel:slot-port-name-type
+--rw portchannel* [name] {fibrechannel:portchannel_platform}?
   ...
```

URI Formats

The URI format for this module takes one of the following forms:

• *<base_URI>*/running/brocade-interface/fibrechannel/ to return a list of all Fibre Channel ports on the specified switch.

• *<base_URI>*/running/brocade-interface/fibrechannel-statistics/ to return the statistics for all Fibre Channel ports on the specified switch.

• *<base_URI>*/running/brocade-interface/fibrechannel-statistics/*fibre-channel-interface-name*/trunk-enabled/ to return the trunk settings for a physical port.

• *<base_URI>*/running/brocade-interface/fibrechannel/name/*fibre-channel-interface-name*/n-port-enabled to configure a port as an N_Port.

• *<base_URI>*/running/brocade-switch/fibrechannel-switch/name/*switch-worldwide-name*/ag-mode to determine if the switch is in Access Gateway mode.

Parameters

brocade-interface

Description: All Fibre Channel interface-related configuration, operational state, and statistics.
Flag: read-write
Type: string
This container has the following leafs:

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.
Flag: read-write
Key: *name*
This list has the following leafs:

name

Description: The name of the interface.
Flag: read-write
Type: string
Values: The slot and port number of the specified port in the format slot/port.
Optional: No

wwn

Description: The world wide name (WWN) of the specified port.
Flag: read-only
Type: fibrechannel:wwn-type
Config: false
Values: A valid WWN name.
Optional: Yes

operational-status

Description: The current operational status of the specified port.
Flag: read-only
Type: uint32
Config: false
Values:
0 = Undefined
2 = Online/Testing
3 = Offline

5 = Faulty
Optional: Yes

enabled-state
Description: The parameter is deprecated. Use the is-enabled-state parameter. The current state of the specified port.
Flag: read-write
Type: uint32
Config: false
Values: 2 = Port is enabled. 6 = Port is disabled.
Optional: Yes

is-enabled-state
Description: The current state of the specified port.
Flag: read-write
Type: boolean
Values: true = Port is enabled. false = Port is disabled.
Optional: Yes

user-friendly-name
Description: A user-friendly name to be assigned to the specified port.
When a port name is not configured, a default name is assigned. The format of the default name is as follows:
• On stand-alone 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.
Flag: read-write
Type: string
Config: false
Values: 1 to 128 alphanumeric characters. This name must not contain special (high-ASCII) characters, and it is not case-sensitive.
Optional: Yes

speed
Description: 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 0 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.
Flag: read-write
Type: speed-type
Values:
0 - Auto-negotiated speed (Default)
1000000000 = Fixed at 1 Gb/s
2000000000 = Fixed at 2 Gb/s
4000000000 = Fixed at 4 Gb/s
8000000000 = Fixed at 8 Gb/s
10000000000 = Fixed at 10 Gb/s
16000000000 = Fixed at 16 Gb/s
32000000000 = Fixed at 32 Gb/s
Optional: Yes

max-speed

Description: The maximum speed the port is capable of supporting in bits per second.
Flag: read-only
Type: speed-type
Config: false
Values:
0 - Auto-negotiated speed (Default)
1000000000 = Fixed at 1 Gb/s
2000000000 = Fixed at 2 Gb/s
4000000000 = Fixed at 4 Gb/s
8000000000 = Fixed at 8 Gb/s
10000000000 = Fixed at 10 Gb/s
16000000000 = Fixed at 16 Gb/s
32000000000 = Fixed at 32 Gb/s
Optional: Yes

auto-negotiate

Description: Whether the port speed is auto-negotiated on the specified port.
Flag: read-only
Type: uint8
Config: false
Values: 0 = Port speed is fixed. **1** = Port speed is auto-negotiated.
Optional: Yes

g-port-locked

Description: Designates the specified port as a G_Port. Note that the Fibre Channel switch Native mode must be supported. 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.
Flag: read-write
Type: uint8
Config: false
Values: 0 = Unlocked (removes the G_Port designation from the port). 1 = Locked (the port is designated as a G_Port).
Optional: Yes

e-port-disable

Description: Enables or disables E_Port capability on the specified port or locks down the specified port as an E_Port. Note that the Fibre Channel switch Native mode must be supported. E_Port capability is enabled by default. When an inter-switch link (ISL) is connected to a port and the port's E_Port capability 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.
Flag: read-write
Type: uint8
Config: false
Values: 0 = Enables the port as an E_Port. **1** = Disables E_Port capability for the specified port. Default: **0**.
Optional: Yes

n-port-enabled

Description: Indicates whether this port can operate as an N_Port. Note that the Fibre Channel switch must be in Access Gateway mode.

Flag: read-write
Type: uint8
Values: 0 = Port is disabled as an N_Port. **1** = Port is enabled as an N_Port.
Optional: Yes

d-port-enable
Description: Indicates whether this port is configured as a D_Port. This resource is only on 16-Gb/s-capable blades that support D_Port capability.
Flag: read-write
Type: uint8
Config: false
Values: 0 = Port is disabled as a D_Port. **1** = Port is enabled as a D_Port.
Optional: Yes

persistent-disable
Description: Status of the persistent-disable feature for the port.
Flag: read-write
Type: uint8
Config: false
Values: 0 = Persistent-disable is not active for the port. **1** = Persistent-disable is active for the port.
Optional: Yes

neighbor
Description: A list of WWNs.
Flag: read-only
Config: false
This container has the following leaf:

wwn
Description: The Fibre Channel WWN of the neighbor port.
Flag: read-only
Type: fibrechannel:wwn-type
Config: false
Values: A valid WWN name.
Optional: No

neighbor-node-wwn
Description: The neighbor node WWN connected to this port. This parameter is available only when the switch operational status is online or testing (operational-status = 2).
Flag: read-only
Type: fibrechannel:wwn-type
Config: false
Values: A valid WWN name.
Optional: No

fcid
Description: This parameter is deprecated. Use the fcid-hex parameter. The Fibre Channel ID (FCID) of the specified port (decimal format).
Flag: read-only
Type: fibrechannel:fcid-type
Config: false
Values: A valid Fibre Channel ID.
Optional: Yes

fcid-hex
Description: The Fibre Channel ID (FCID) of the specified port (hexidecimal format).

Flag: read-only
Type: fibrechannel:fcid-hex-string-type
Config: false
Values: A valid Fibre Channel ID of the port.
Optional: Yes

port-type
Description: 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.
Flag: read-only
Type: port-type-type
Config: false
Values:
0 = Unknown
7 = E_Port
10 = G_Port
11 = U_Port (Default)
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
32768 = LB_Port
Optional: Yes

port-type-string
Description: The currently enabled port type for the specified port. If the port is logged in, it displays the negotiated port type; otherwise, it displays the configured port type.
Flag: read-only
Type: port-type-string-type
Config: false
Value: e-port = The port is an E_Port. g-port = The port is a G_Port. universal-port = The port is not configured for any type. f-port = The port is a F_Port. l-port = The port is a Logical port. fcoe-port = The port is a FCoE_Port. ex-port = The port is a EX_Port. d-port = The port is a Diagnostic port. sim-port = The port is SIM port. af-port = The port is AF_Port. AF_Ports are present only on AMP switches. ae-port = The port is an AE_Port. ve-port = The port is a VE_Port. ethernet-port = The port is a Ethernet Flex port. flex-port = The port is a Flex port. n-port = The port is N_Port. N_Ports are applicable for AG mode. mirror-port = The port is a Mirror port. encryption-support-port = The port is an Encryption Support Port. A front-end port that is reserved for the Encryption block to work. loopback-port = The port is a Loopback port. unknown-port = The port is online and does not belong any known port type.
Optional: No

qos-enabled
Description: Indicates whether QoS is enabled on the port.
Flag: read-write
Type: uint8
Config: false

Values: 0 = Port QoS disabled. **1** = Port QoS enabled.
Optional: Yes

compression-configured
Description: 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 number of configurable ports is limited by the platform ASIC. The switch must be an FC switch in Native mode.
Flag: read-write
Type: uint8
Config: false
Values: 0 = Compression configuration disabled. **1** = Compression configuration enabled.
Optional: Yes

encryption-enabled
Description: Enables or disables the encryption configuration on the port. This change is persistent. 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 number of configurable ports is limited by the platform ASIC. The switch must be an FC switch in Native mode.
Flag: read-write
Type: uint8
Value: 0 - Disables the encryption configuration on the specified port. 1 - Enables the encryption configuration on the specified port.
Optional: No

compression-active
Description: The runtime compression status for the specified port. The switch must be an FC switch in Native mode.
Flag: read-only
Type: uint8
Config: false
Values: 0 = Port is enabled for compression but offline, or not enabled for compression. **1** = Port is online and compression is enabled.
Optional: Yes

encryption-active
Description: The encryption status for the specified port. The switch must be an FC switch in Native mode.
Flag: read-only
Type: uint8
Config: false
Values: 0 = Encryption is disabled. **1** = Encryption is enabled.
Optional: Yes

target-driven-zoning-enable
Description: Enables or disables the Target Driven Zoning (TDZ) configuration on the specified port. This change is persistent. Target Driven Zoning 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 Target Driven Zoning is configured on a port, it allows the connected target device to configure Target Driven Peer Zones to be enabled and committed.
Flag: read-write
Type: uint8
Config: false

Values: 0 = Target Driven Zoning configuration is disabled. **1** = Target Driven Zoning configuration is enabled.
Optional: Yes

sim-port-enabled
Description: Enables or disables the port as a SIM port. Note that Fibre Channel switch Native mode must be supported.
Flag: read-write
Type: uint8
Values: 0 = SIM port is disabled. **1** = SIM port is enabled.
Optional: Yes

mirror-port-enabled
Description:
Enables or disables the port as a mirror port. Port mirroring reroutes data frames between two devices to the mirror port.
Flag: read-write
Type: int8
Values: 0 = Mirror port is disabled. **1** = Mirror port is enabled.
Optional: Yes

credit-recovery-enabled
Description: Enables or disables credit recovery on the port. Credit recovery enables credits or frames to be recovered.
Flag: read-write
Type: int8
Values: 0 = Credit recovery is disabled. **1** = Credit recovery is enabled. Default: **1**.
Optional: Yes

credit-recovery-active
Description: 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.
Flag: read-only
Type: int8
Config: false
Values: 0 = Credit recovery is not active on the port. **1** = Credit recovery is active on the port.
Optional: Yes

fec-active
Description: 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.
Flag: read-only
Type: int8
Config: false
Values: 0 = FEC is not active on the port. **1** = FEC is active on the port.
Optional: Yes

f-port-buffers
Description: 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.
Flag: read-write
Type: uint8
Values: 0 or from 8 to the maximum.

Optional: Yes

e-port-credit

Description: 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.
Flag: read-write
Type: uint8
Values: 0 or from 5 to 160. **0** = E_Port Credit is disabled.
Optional: Yes

csctl-mode-enabled

Description: Enables or disables Class-Specific Control (CSCTL) mode which enables traffic prioritization based on CS_CTL. The switch must be an FC switch in Native mode.
Flag: read-write
Type: int8
Values: 0 = CSCTL mode is disabled on the port. **1** = CSCTL mode is enabled on the port.
Optional: Yes

fault-delay-enabled

Description: 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.
Flag: read-write
Type: uint8
Values: 0 = The value is R_A_TOV. **1** = The value is 1.2 seconds. Default: **0**.
Optional: Yes

octet-speed-combo

Description: The speed configuration for a port octet. 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.
Flag: read-write
Type: int8
Values:
1 = Auto-negotiated or fixed port speeds of 32 Gb/s, 16 Gb/s, 8 Gb/s, 4 Gb/s, and 2 Gb/s.
2 = Auto-negotiated or fixed port speeds of 10 Gb/s, 8 Gb/s, 4 Gb/s, and 2 Gb/s.
3 = Auto-negotiated or fixed port speeds of 16 Gb/s and 10 Gb/s.
Default = **1**.
Optional: Yes

long-distance

Description: The long-distance level. Note that Fibre Channel switch Native mode must be supported.
Flag: read-only
Type: uint8
Config: false
Values: 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 value as long (<= 50 km)
3 - L2 configures the value as super long (<= 100 km)
4 - LE mode configures an E_Port distance as greater than 5 km and up to 10 km.
5 - L0.5 configures the value 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.
Optional: Yes

vc-link-init
Description: The VC link initialization. Note that Fibre Channel switch Native mode must be supported.
Flag: read-only
Type: uint8
Config: false
Values: 0 = The long-distance link initialization is turned off. **1** = The long-distance link initialization is turned on for long-distance mode.
Optional: Yes

isl-ready-mode-enabled
Description: Enables or disables ISL ready mode on the port. Note that Fibre Channel switch Native mode must be supported. Note that ISL ready mode is mutually exclusive with E_Port Credit, QoS-enabled mode, and long-distance levels.
Flag: read-write
Type: uint8
Values: 0 = ISL ready mode is disabled on the port. **1** = ISL ready mode is enabled on the port.
Optional: Yes

rscn-suppression-enabled
Description: Enables or disables Registered State Change Notification (RSCN) suppression on the port. Note that Fibre Channel switch Native mode must be supported.
When enabled, any device changes on the port do not generate an RSCN to any other end device. 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.
Flag: read-write
Type: uint8
Values: 0 = RSCN is disabled on the port. **1** = RSCN is enabled on the port. Default: **0**.
Optional: Yes

los-tov-mode-enabled
Description: Enables or disables de-bouncing of signal loss for front-end ports for 100 ms.
Flag: read-write
Type: uint8
Values: 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 for both fixed-speed and auto-negotiated ports.
Optional: Yes

npiv-enabled
Description: Enables or disables NPIV capability on the port.
Flag: read-write
Type: uint8
Values: 0 = NPIV is disabled on the port. **1** = NPIV is enabled on the port.
Optional: Yes

npiv-pp-limit
Description: Configures the maximum number of allowed logins for the port.
Flag: read-write
Type: int16
Values: 1 to **255** = The maximum number of allowed logins for the port. Default = 126.
Optional: Yes

npiv-flogi-logout-enabled

Description: Whether or not base device logout is enabled or disabled.
Flag: read-only
Type: int8
Config: false
Values: 0 = Base device logout is disabled on the port, which causes 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.
Optional: Yes

ex-port-enabled

Description: Configures the port as an EX_Port. Note that Fibre Channel switch Native mode must be supported.
Flag: read-write
Type: uint8
Values: 0 = Does not configure the port as an EX_Port. **1** = Configures the port as an EX_Port.
Optional: Yes

edge-fabric-id

Description: The fabric ID. This parameter is applicable only when the EX_Port is enabled (ex-port-enabled = 1).
Flag: read-write
Type: fibrechannel:fabric-id-type
Value: 1 to 128.
Optional: No

preferred-front-domain-id

Description: The preferred domain ID. This parameter is applicable only when the EX_Port is enabled (ex-port-enabled = 1).
Flag: read-write
Type: fibrechannel:domain-id-type
Value: 1 to 239.
Optional: No

neighbor-switch-ipv4-address

Description: The IPv4 address of the switch connected to the EX-port. For trunked EX_Ports, this IP address is available only on the master port. This parameter is applicable only when the EX_Port is enabled (ex-port-enabled = 1).
Flag: read-only
Type: inet:ipv4-address
Config: false
Value: An IPv4 address.
Optional: No

neighbor-switch-ipv6-address

Description: The IPv6 address of the switch connected to the EX-port. For trunked EX_Ports, this IP address is available only on the master port. This parameter is applicable only when the EX_Port is enabled (ex-port-enabled = 1).
Flag: read-only
Type: inet:ipv6-address
Value: An IPv6 address.
Optional: No

fec-enabled

Description: 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 Gb/s 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. **Note:** FEC only supported at 10 Gb/s and 16 Gb/s speeds.
Flag: read-write
Type: int8
Values: 0 = FEC is disabled on the port. **1** = FEC is enabled on the port, and the port is online.
Optional: Yes

via-tts-fec-enabled
Description: Whether FEC negotiation via TTS is enabled or disabled on the port.
Flag: read-only
Type: int8
Config: false
Values: 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.
Optional: Yes

port-autodisable-enabled
Description: Enables or disables port autodisable on the port, which minimizes traffic disruption introduced in some instances of automatic port recovery. 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.
Flag: read-write
Type: uint8
Values: 0 = Port autodisable is disabled on the port. **1** = Port autodisable is enabled on the port.
Optional: Yes

rate-limit-enabled
Description: This parameter is obsolete. Configures the rate limits in Mb/s. Note that Fibre Channel switch Native mode must be supported. 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.
Flag: read-write
Type: int16
Values: One of the following rates in Mb/s: 0 (no rate limit set), 200, 400, 600, 800, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 5000, 7000, 8000, 9000, 10000, 11000, 12000, 13000, 14000, 15000, 16000.
Optional: Yes

non-dfe-enabled
Description: This parameter is obsolete. Enables or disables non-DFE mode on the port. Note that Fibre Channel switch Native mode must be supported. Active receiver DFE uses sophisticated algorithms to automatically adjust the receiver to compensate for signal distortions.
Flag: read-write
Type: uint8
Values: 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 the following conditions are met: the port speed is 8G or N8 and the received fillword is IDLE.
Optional: Yes

trunk-port-enabled
Description: Enables or disables trunking on a port. Trunking is enabled by default when a trunking license is present on the switch.
Flag: read-write
Type: uint8
Values: 0 = Trunking is disabled on the port. **1** = Trunking is enabled on the port.
Optional: Yes

default-index
Description: The default port Index of the port. The default port index can be a port swapped area.
Flag: read-write
Type: fibrechannel:user-port-number-type
Config: false
Values: The default port Index of the port.
Optional: Yes

physical-state
Description: The physical state of a port.
Flag: read-only
Type: string
Config: false
Values: The physical state of a port (online|offline|testing|faulty|e_port|f_port|segmented|unknown|no_port|no_module|laser_flt|no_light|no_sync|in_sync|port_flt|hard_flt|diag_flt|lock_ref|mod_inv|mod_val|no_sigdet).
Optional: Yes

pod-license-status
Description: This parameter is deprecated. Use the pod-license-state parameter instead. The POD license status for a port.
Flag: read-only
Type: boolean
Config: false
Values: true = The POD license is enabled on the port. **false** = The POD license is disabled on the port.
Optional: Yes

pod-license-state
Description: The Ports on Demand (POD) license status of the logical port. The switch must be a non-director FC switch.
Flag: read-write
Type: enumeration
Value: reserved = The port is reserved under a POD license. released = The port is not reserved under a POD license.
Optional: No

index
Description: The user port number of the front-end port.
Flag: read-only
Type: fibrechannel:user-port-number-type
Config: false
Value: -1 to 3400.
Optional: No

reserved-buffers

Description: The maximum number of buffers or reserved buffers if offline. This value increases or decreases when you configure the f-port-buffers parameter and is effective when the port is online. If the port is offline, the default value is assigned. The maximum value is derived from the chip-buffers-available parameter. Note that if the value set is too high, it depletes the buffers available to other front end ports on that chip.
This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port and the reserved-buffers is greater than zero.
Flag: read-only
Type: uint16
Config: false
Value: 1 to maximum number of buffers or reserved buffers if offline.
Optional: No

average-transmit-buffer-usage

Description: The average number of buffers for transmit per polling period.
This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port and the average-transmit-buffer-usage is greater than zero.
Flag: read-only
Type: uint16
Config: false
Value: The average number of buffers for transmit per polling period.
Optional: No

average-transmit-frame-size

Description: The average frame size for transmit, including the FC header, in bytes.
This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port and the average-transmit-frame-size is greater than zero.
Flag: read-only
Type: uint16
Config: false
Value: The average frame size for transmit, including the FC header, in bytes.
Optional: No

average-receive-buffer-usage

Description: The average number of buffers for receive per polling period.
This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port and the average-receive-buffer-usage is greater than zero.
Flag: read-only
Type: uint16
Config: false
Value: The average number of buffers for receive per polling period.
Optional: No

average-receive-frame-size

Description: The average frame size for receive, including the FC header, in bytes.
This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port and the average-receive-frame-size is greater than zero.
Flag: read-only
Type: uint16
Config: false
Value: The average frame size for receive, including the FC header, in bytes.
Optional: No

current-buffer-usage

Description: The real-time buffer usage by a front end port.

This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port and the current-buffer-usage is greater than zero.
Flag: read-only
Type: uint16
Config: false
Value: The real-time buffer usage by a front end port.
Optional: No

recommended-buffers
Description: The real-time number of buffers needed to utilize the port at full bandwidth. If this value is too low, the port starves for buffers. If this value is too high, buffers are wasted and not available to other front end ports on the same chip.
This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port and the recommended-buffers is greater than zero..
Flag: read-only
Type: uint16
Config: false
Value:
Optional: No

measured-link-distance
Description: The actual link distance measured in kilometers.
This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port.
Flag: read-only
Type: string
Config: false
Value: 1 to 8 kilometers.
Optional: No

chip-instance
Description: The chip index on which the given port exists.
This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port.
Flag: read-only
Type: uint16
Config: false
Value: The chip index on which the given port exists.
Optional: No

chip-buffers-available
Description: The front end port on the chip whose available buffer count is being queried.
This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port.
Flag: read-only
Type: uint16
Config: false
Value:
Optional: No

port-health
Description: The port health (for example, Offline).
Flag: read-only
Type: string
Config: false
Value: The port health (for example, Offline).

Optional: No

authentication-protocol
Description: Whether authentication is enabled and the authentication protocol configured on the port.
Flag: read-only
Type: string
Config: false
Value: Whether authentication is enabled and the authentication protocol configured on the port.
Optional: No

disable-reason
Description: The reason the port is disabled. Use a PATCH request with an empty string to clear the user-defined reason. You can also enable an offline port to clear the user-defined reason.
Flag: read-write
Type: string
Value: The reason the port is disabled.
Optional: No

areas
Description: A list of 16-bit areas associated with a Gen7 F_Port.
Flag: read-only
Config: false
Optional: No

area
Description: The area of a device.
Flag: read-only
Type: fibrechannel:fcid-hex-string-type
Value: The area of a device.
Optional: No

le-domain
Description: The LE domain ID for the port.
Flag: read-only
Type: uint16
Value: 1 to 128.
Optional: No

port-peer-beacon-enabled
Description: Enables or disables port peer beaconing.
Flag: read-write
Type: boolean
Value: true = Enables port peer beaconing. false = Disables port peer beaconing.
Optional: No

clean-address-enabled
Description: Enables or disables Clean Address Bit on the specified port. The switch must be a non-director FC switch.
Flag: read-write
Type: boolean
Value: true : Enables Clean Address Bit support for the specified port. false : Disables Clean Address Bit support for the specified port.
Optional: No

congestion-signal-enabled? boolean {fibrechannel:fibrechannel_switch_platform}?

Description: Enables or disables Congestion Signal support for the specified port. Only available on Gen7 or later platforms.
This parameter is available only when the port type string is not an FCoE_Port, VE_Port, and Ethernet port.
Flag: read-write
Type: boolean
Value: true = Enables Congestion Signal support for the specified port. false = Disables Congestion Signal support for the specified port.
Optional: No

segmentation-reason

Description: Displays the reason for port segmentation.
Flag: read-only
Type: string
Config: false
Value: Displays the reason for port segmentation.
Optional: No

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.
Flag: read-write
Key: *name*
This list has the following leafs:

name

Description: The name of the interface.
Flag: read-write
Type: string
Config: false
Values: The slot and port number of the port in the format slot/port.
Optional: No

sampling-interval

Description: The sampling interval for statistics.
Flag: read-only
Type: uint16
Config: false
Units: seconds
Values: 1 through 65536.
Optional: Yes

time-generated

Description: The time at which the statistics were queried.
Flag: read-only
Type: fibrechannel:time-generated-type
Config: false
Units: seconds
Values: standard IETF date-time output
Optional: Yes

reset-statistics

Description: Resets statistic counters. The only readable value is 0, and the only settable value is 1. This is a write only parameter.
Flag: read-write

Type: uint8
Config: false
Values: 0 = Do not reset the statistic counters. 1 = Reset the statistic counters.
Optional: Yes

in-octets
Description: The total number of octets received on the interface, including framing characters.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: octet count
Values: 0 through 18446744073709551615
Optional: Yes

out-octets
Description: The total number of octets transmitted out of the interface, including framing characters.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: octet count
Values: 0 through 18446744073709551615
Optional: Yes

in-multicast-pkts
Description: 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.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: packet count
Values: 0 through 18446744073709551615
Optional: Yes

out-multicast-pkts
Description: 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.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: packet count
Values: 0 through 18446744073709551615
Optional: Yes

in-link-resets
Description: The number of link resets received.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

out-link-resets
Description: The total number of link resets transmitted.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

in-offline-sequences
Description: The total number of offline sequences received.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

out-offline-sequences
Description: The total number of offline sequences transmitted.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

invalid-ordered-sets
Description: The total number of invalid ordered sets received.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: set count
Values: 0 through 18446744073709551615
Optional: Yes

frames-too-long
Description: 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.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

truncated-frames
Description: The total number of truncated frames received.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

address-errors

Description: Count of frames received with unknown addressing. An example is an unknown SID or DID, which are not known to the routing algorithm.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

delimiter-errors

Description: Count of invalid frame delimiters that are received at this port. An example would be a frame that has a class 2 at the start and a class 3 at the end.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

encoding-disparity-errors

Description: The total number of disparity errors received at this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: error count
Values: 0 through 18446744073709551615
Optional: Yes

too-many-rdys

Description: The number of instances in which the number of RDYs (readys) exceeded the number of frames received.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

in-crc-errors

Description: The number of CRC errors for all frames received.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: error count
Values: 0 through 18446744073709551615
Optional: Yes

crc-errors

Description: 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).
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: error count
Values: 0 through 18446744073709551615

Optional: Yes

bad-eofs-received
Description: The number of bad EOF frames received.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

encoding-errors-outside-frame
Description: The number of encoding-error or disparity-error outside frames received.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

multicast-timeouts
Description: The number of multicast frames that have timed out.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

in-lcs
Description: The number of link control (lcs) frames received.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

in-frame-rate
Description: The instantaneous receive frame rate.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

out-frame-rate
Description: The instantaneous transmit frame rate.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

in-max-frame-rate

Description: The maximum frame receive rate since the last reset.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

out-max-frame-rate

Description: The maximum frame transmit rate since the last reset.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

in-rate

Description: The instantaneous byte receive rate.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: bytes
Values: 0 through 18446744073709551615
Optional: Yes

out-rate

Description: The instantaneous byte transmit rate.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: bytes
Values: 0 through 18446744073709551615
Optional: Yes

in-peak-rate

Description: The peak byte receive rate.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: bytes
Values: 0 through 18446744073709551615
Optional: Yes

out-peak-rate

Description: The peak byte transmit rate.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: bytes
Values: 0 through 18446744073709551615
Optional: Yes

in-frames

Description: The number of frames received at this port.

Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

out-frames
Description: The number of frames transmitted from this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

bb-credit-zero
Description: The number of transitions in and out of the BB credit zero state.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: transition count
Values: 0 through 18446744073709551615
Optional: Yes

input-buffer-full
Description: The number of transitions in and out of the Input Buffer Full state.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: transition count
Values: 0 through 18446744073709551615
Optional: Yes

f-busy-frames
Description: The number of F_BSY (fabric busy) frames generated.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

p-busy-frames
Description: The number of P_BSY (port busy) frames generated.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

f-rjt-frames
Description: The number of F_RJT (fabric frame reject) frames generated.
Flag: read-only
Type: yang:zero-based-counter64

Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

p-rjt-frames
Description: The number of P_RJT (port frame reject) frames generated.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

class-1-frames
Description: The number of Class 1 frames received at this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

class-2-frames
Description: The number of Class 2 frames received at this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

class-3-frames
Description: The number of Class 3 frames received at this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

class-3-discards
Description: The number of Class 3 frames discarded by this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

link-failures
Description: The number of link failures at this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count

Values: 0 through 18446744073709551615
Optional: Yes

invalid-transmission-words
Description: The number of invalid transmission words received at this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: word count
Values: 0 through 18446744073709551615
Optional: Yes

primitive-sequence-protocol-error
Description: The number of primitive sequence protocol errors detected at this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: error count
Values: 0 through 18446744073709551615
Optional: Yes

loss-of-signal
Description: The number of signal loss instances detected at this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

loss-of-sync
Description: The number of instances of synchronization loss detected at this port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

class3-in-discards
Description: The number of class 3 receive frames discarded due to timeout.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

class3-out-discards
Description: The number of class 3 transmit frames discarded due to timeout.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

pcs-block-errors

Description: The number of physical coding sublayer (PCS) block errors. This counter records encoding violations on 10-Gb/s or 16-Gb/s ports.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: error count
Values: 0 through 18446744073709551615
Optional: Yes

remote-link-failures

Description: The number of link failures at the remote F-port. This parameter is available only when the swtich is online (/fibrechannel/operational-status = 2).
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

remote-invalid-transmission-words

Description: The number of invalid transmission words received at the remote F_Port. This parameter is available only when the swtich is online (/fibrechannel/operational-status = 2).
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: word count
Values: 0 through 18446744073709551615
Optional: Yes

remote-primitive-sequence-protocol-error

Description: The number of primitive sequence protocol errors detected at the remote F_Port. This parameter is available only when the swtich is online (/fibrechannel/operational-status = 2).
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: errors count
Values: 0 through 18446744073709551615
Optional: Yes

remote-loss-of-signal

Description: The number of instances of signal loss detected at the remote F_Port. This parameter is available only when the swtich is online (/fibrechannel/operational-status = 2).
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

remote-loss-of-sync

Description: The number of instances of synchronization loss detected at the remote F_Port. This parameter is available only when the swtich is online (/fibrechannel/operational-status = 2).
Flag: read-only
Type: yang:zero-based-counter64
Config: false

Units: instance count
Values: 0 through 18446744073709551615
Optional: Yes

remote-crc-errors
Description: The number of frames received with invalid CRC at the remote F_Port. This parameter is available only when the swtich is online (/fibrechannel/operational-status = 2).
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

remote-fec-uncorrected
Description: The number of frames uncorrected by the FEC block at the remote F_Port. This parameter is available only when the swtich is online (/fibrechannel/operational-status = 2).
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: frame count
Values: 0 through 18446744073709551615
Optional: Yes

remote-buffer-credit-info
Description: The buffer credit values. Applicable only for the remote F_Port. This parameter is available only when the swtich is online (/fibrechannel/operational-status = 2).
Flag: read-only
Type: yang:zero-based-counter64
Config: false
The container has the following leafs:

bb-credit
Description: The buffer credit values available to the attached devices. This parameter is available only when the swtich is online (/fibrechannel/operational-status = 2).
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: transition count
Values: 0 through 18446744073709551615
Optional: Yes

peer-bb-credit
Description: The number of credits available to the switch port. This parameter is available only when the swtich is online (/fibrechannel/operational-status = 2).
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Units: transition count
Values: 0 through 18446744073709551615
Optional: Yes

link-level-interrupts
Description: The number of link level interrupts on the port.
Flag: read-only
Type: yang:zero-based-counter64

Config: false
Value: The number of link level interrupts on the port.
Optional: No

frames-processing-required
Description: The number of frames which required processing on the port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Value: The number of frames which required processing on the port.
Optional: No

frames-timed-out
Description: The number of frames which timed out during transmit on the port.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Value: The number of frames which timed out during transmit on the port.
Optional: No

frames-transmitter-unavailable-errors
Description: The number of frames returned by an unavailable transmitter.
Flag: read-only
Type: yang:zero-based-counter64
Config: false
Value: The number of frames returned by an unavailable transmitter.
Optional: No

logical-e-port
Description: A list of logical E_Port interfaces on the device which form the logical interswitch link (LISL). The switch must support Extended ISL (XISL) and Virtual Fabrics mode must be enabled.
Flag: read-only
Key: port-index
Optional: No

port-index
Description: The unique port number on the switch that identifies a logical E_Port.
Flag: read-only
Type: fibrechannel:user-port-number-type
Value: The logical E_Port user port number on the switch.
Optional: No

fabric-id
Description: The virtual fabric ID (VFID) of the logical switch on which the logical E_Port was created.
Flag: read-only
Type: fibrechannel:fabric-id-type
Value: The VFID of the logical switch for the logical E_Port.
Optional: No

operational-status
Description: The operational status of the the logical E_Port.
Flag: read-only
Type: enumeration
Value: online = The logical E-Port is online. offline = The logical E-Port is offline.
Optional: No

offline-reason

Description: The reason for the logical E_Port offline status.
Flag: read-only
Type: enumeration
Value: Init = The logical E_Port is initializing. CreateRequestSent = The request to create a logical E_Port is sent to the peer switch. CreatReqRtryPendg = Retry pending to send a create request to the peer switch. PeerLSDown = Peer switch is disabled or offline. CreateFailed = The logical E_Port creation failed. PeerReqInProcess = The request from the peer switch to create a logical E_Port is in process. PeerReqReceived = The request from the peer switch to create a logical E_Port is received. DeleteSent = The request to delete the logical E_Port sent to the peer switch. Invalid = The logical E_Port is in an Invalid state.
Optional: No

neighbor-node-wwn

Description: The neighbor node WWN logically connected to this port.
Flag: read-only
Type: fibrechannel:wwn-type
Value: The neighbor node WWN logically connected to this port.
Optional: No

associated-physical-ports

Description: A list of associated physical E_Ports.
Flag: read-only
Optional: No
This list has the following leaf:

port

Description: The physical port number of the base switch associated with the logical E_Port.
Flag: read-only
Type: fibrechannel:slot-port-name-type
Value: The physical port number of the base switch associated with the logical E_Port.
Optional: No

Supported Methods

Only the GET, HEAD, OPTIONS, and PATCH operations are supported in this module.

Examples

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 Fibre Channel Resources

This example uses a GET request to retrieve a list of all Fibre Channel resources in the fabric.

Structure

GET *<base_URI>*/running/brocade-interface/fibrechannel

URI

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

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>0/0</name>
      <wwn>10:10:10:eb:1a:b7:77:bc</wwn>
      <port-type>11</port-type>
      <speed>16000000000</speed>
      <user-friendly-name>port0</user-friendly-name>
      <operational-status>3</operational-status>
      <is-enabled-state>2</is-enabled-state>
      <auto-negotiate>1</auto-negotiate>
      <fcid>65536</fcid>
      <persistent-disable>0</persistent-disable>
      <g-port-locked>0</g-port-locked>
      <e-port-disable>0</e-port-disable>
      <qos-enabled>0</qos-enabled>
      <d-port-enable>0</d-port-enable>
      <compression-configured>0</compression-configured>
      <compression-active>0</compression-active>
      <encryption-active>0</encryption-active>
      <neighbor/>
      <target-driven-zoning-enable>0</target-driven-zoning-enable>
   </fibrechannel>
    .
    .
    .
   <fibrechannel>
      <name>0/47</name>
      <wwn>10:1f:10:eb:1a:b7:77:bc</wwn>
      <port-type>11</port-type>
      <speed>16000000000</speed>
      <user-friendly-name>port47</user-friendly-name>
      <operational-status>3</operational-status>
      <is-enabled-state>6</is-enabled-state>
      <auto-negotiate>1</auto-negotiate>
      <fcid>77568</fcid>
      <persistent-disable>0</persistent-disable>
      <g-port-locked>0</g-port-locked>
      <e-port-disable>0</e-port-disable>
      <qos-enabled>0</qos-enabled>
      <d-port-enable>0</d-port-enable>
      <compression-configured>0</compression-configured>
      <compression-active>0</compression-active>
      <encryption-active>0</encryption-active>
      <neighbor/>
      <target-driven-zoning-enable>0</target-driven-zoning-enable>
   </fibrechannel>
</Response>
```

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 0/10.

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/0%2f10
```

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>0/10</name>
    <wwn>10:1a:14:15:1c:9e:3b:aa</wwn>
    <port-type>11</port-type>
    <speed>32000000000</speed>
    <user-friendly-name>port10</user-friendly-name>
    <operational-status>3</operational-status>
    <is-enabled-state>6</is-enabled-state>
    <fcid>68096</fcid>
    <persistent-disable>0</persistent-disable>
    <g-port-locked>0</g-port-locked>
    <e-port-disable>0</e-port-disable>
    <qos-capabilities>3</qos-capabilities>
    <qos-enabled>1</qos-enabled>
    <d-port-enable>0</d-port-enable>
    <compression-configured>0</compression-configured>
    <compression-active>0</compression-active>
    <encryption-active>0</encryption-active>
    <neighbor/>
  </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/*fibre-channel-interface-name*/is-enabled-state/*enabled-state*/6

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/6
```

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-enabled>0</trunk-enabled>

  </fibrechannel>
```

```
</Response>
```

History

Release Version	History
Fabric OS 8.2.0	This API call was introduced.
Fabric OS 8.2.0a	This API call was modified to add new parameters to the brocade-interface/fibrechannel container.
Fabric OS 8.2.1	This API call was modified to add n-port-enable, max-speed, and trunk-enable to the brocade-interface/fibrechannel container as well as add new RDP statistic parameters to the brocade-interface/fibrechannel-statistics container.

brocade-license

This module provides a detailed view of the licenses installed on the switch.

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
```

URI format

The URI format for this module takes the following form:

- *<base_URI>*`/running/brocade-license/license` followed by the leafs as listed in the module tree.

Parameters

brocade-license

Description: The container for licenses installed on the switch.
Flag: read-only
Config: false
This container has the following leafs:

license

Description: A list of licenses installed on the switch.
Flag: read-only
Key: *name*
This list has the following leafs:

name

Description: The license key for one or more features installed on the switch.
Flag: read-only
Type: string
Value: The license key.
Optional: Yes

features

Description: A list of features that are integrated in a single license key.
Flag: read-only
This container has the following leaf:

feature

Description: The name of the feature.
Flag: read-only

Type: string
Value: The name of the feature.
Optional: Yes

capacity
Description: The capacity for the license installed on the switch. Note that this parameter is valid only for a capacity-based license.
Flag: read-only
Type: uint32
Value: The capacity for the license installed on the switch.
Optional: Yes

consumed
Description: The number of slots configured to use the license installed on the switch. Note that this parameter is valid only for a capacity-based license.
Flag: read-only
Type: uint32
Value: The number of slots consumed.
Optional: Yes

configured-blade-slots
Description: A list of slot numbers of the configured blade slots for the license installed on the switch.
Flag: read-only
This container has the following leaf:

configured-blade-slot
Description: The configured blade slot details.
Flag: read-only
Type: uint32
Value: The configured blade slot details.
Optional: Yes

expiration-date
Description: The expiration date for the license installed on the switch.
Flag: read-only
Type: string
Value: The expiration date for the license in MM/DD/YYYY format.
Optional: Yes

Supported methods

Only the GET, HEAD, OPTIONS operation is supported in this module.

Examples

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>LWRAXQgAPEAggZtQ44ERFTNGXmmXYYYKBSRHB</name>
        <features>
            <feature>Trunking</feature>
        </features>
    </license>
    <license>
        <name>JA4XFWAWEWLTJr7SMBKf9D3Ga7PmPP37BJZDN</name>
        <features>
            <feature>Fabric Vision</feature>
        </features>
    </license>
    <license>
        <name>KTWFCm3PYafrCCJSfLBWMtWSmW9MY7rCAX7E49FAARWA</name>
        <features>
            <feature>Ports on Demand</feature>
        </features>
        <capacity>16</capacity>
    </license>
    <license>
        <name>tCAQCfDZSgQKrKDENJ9C4KtJaHPgL7TN4XagHAFAmNBB</name>
        <features>
            <feature>Integrated Routing Ports on Demand</feature>
        </features>
        <capacity>8</capacity>
    </license>
    <license>
        <name>CgTfZEAWSF74ARC9CQRJYBSRKRYKKNFTC9TKBmB7Q93DWA7ErF</name>
        <features>
            <feature>Extended Fabric</feature>
        </features>
        <expiration-date>02/18/2019</expiration-date>
    </license>
    <license>
        <name>ZC4LLQDQWFWKFRgfgCDGPZYXLEBWE9DtSDWtLrPQ7YBHgAS4rE</name>
        <features>
            <feature>FICON_CUP</feature>
        </features>
        <expiration-date>11/21/2018</expiration-date>
    </license>
</Response>
```

History

Release version	History
Fabric OS 8.2.1b	This API call was introduced.

brocade-logging

This module provides a detailed view of audit and RASlog message configuration.

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
       |  +--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 keep-alive-period?         uint16
```

URI format

The URI format for this module takes the following form:

• *<base_URI>*`/running/brocade-logging/audit` followed by the leafs as listed in the module tree to display the audit log configuration or to configure audit logging.

• *<base_URI>*`/running/brocade-logging/syslog-server` followed by the leafs as listed in the module tree to display the current syslog server or to configure a syslog server .

• *<base_URI>*`/running/brocade-logging/raslog` followed by the leafs as listed in the module tree to display all external RASlog messages, their status (enabled or disabled), their configured severity and their default severity or configure RASlog messages.

• *<base_URI>*/running/brocade-logging/raslog-module followed by the leafs as listed in the module tree to display a list of all Fabric OS RASlog modules or enable or disable a RASlog module group.

• *<base_URI>*/running/brocade-logging/log-quiet-control followed by the leafs as listed in the module tree to display the quiet time configuration details for audit and RASlog message types or to configure quiet time.

• *<base_URI>*/running/brocade-logging/log-setting followed by the leafs as listed in the module tree to display the syslog facility level and keep alive period or to configure log settings.

Parameters

brocade-logging

Description: The top-level container for all logging related configuration parameters.
Flag: read-write
This container has the following leafs:

brocade-logging

Description: All .
Flag: read-write
This container has the following leafs:

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.
Flag: read-write
This container has the following leafs:

audit-enabled

Description: Enables or disables the audit filters. Note that this does not change the log class and log severity configuration.
Flag: read-write
Type: boolean
Value: true = The audit filter is enabled. **false** = The audit filter is disabled.

severity-level

Description: The severity level of the log messages you want to display. You can set this parameter to display log messages of the specified severity level and higher. You cannot enter multiple severity levels.
Flag: read-write
Type: brocade-logging-types:severity-level-type
Value: info = Displays log messages of info level and higher. **warning** = Displays log messages of warning level and higher. **error** = Displays log messages of error level and higher. **critical** = Displays log messages of critical level and higher.

filter-class-list

Description: A list of class types needed for audit configuration.
Flag: read-write
This container has the following leaf:

filter-class

Description: The filters to be configured for audit classes. You can set more than one audit class, separated by a comma.
Flag: read-write
Type: brocade-logging-types:class-type
Value: zone = Displays the zone audit class. **security** = Displays the security audit class. **configuration** = Displays the configuration audit class. **firmware** = Displays the firmware

audit class. **fabric** = Displays the fabric audit class. **ls** = Displays the ls audit class. **cli** = Displays the cli audit class. **maps** = Displays the maps audit class.

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.
Flag: read-write
Key: server
This list has the following leafs:

server

Description: The IPv4 or IPv6 address or DNS name of the server.
Flag: read-write
Type: inet:host
Value: A valid IPv4 or IPv6 address or DNS name of the server.
Optional: No

port

Description: The target syslog server's TCP/IP port number. The port can only be specified (non-default) when secure mode is enabled on the server (secure-mode=true).
Flag: read-write
Type: inet:port-number
Value: A valid TCP/IP port number. **514** = The default port number for non-secure mode. **6514** = The default port number for secure mode.
Optional: Yes

secure-mode

Description: Whether secure syslog mode to send the error log messages securely using the TLS protocol to the syslog server is enabled.
Flag: read-write
Type: boolean
Value: true = Secure mode is enabled. **false** = Secure mode is disabled. The default is disabled (false).
Optional: Yes

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.
Flag: read-write
Key: message-id
This list has the following leafs:

message-id

Description: The unique identifier for each RASlog message.
Flag: read-write
Type: brocade-logging-types:message-id-type
Value: The unique identifier for the RASlog message.

message-enabled

Description: Whether message logging is enabled for RASlog messages identified by message ID.
Flag: read-write
Type: boolean
Value: true = Message logging is enabled. **false** = Message logging is disabled.

message-flooded

Description: Whether the RASlog message is blocked due to flooding.
Flag: read-only
Type: boolean
Value: true = The message is blocked due to flooding. **false** = The message is not blocked due to flooding.

syslog-enabled

Description: Whether internal RASlog messages to be sent to syslog is enabled. This parameter is only supported for internal RASlog messages. Internal RASlog messages have an ID number equal to or greater than 5000.
Flag: read-write
Type: boolean
Value: true = Syslog is enabled. **false** = Syslog is disabled.

message-text

Description: The message text corresponding to the internal RASlog message. This parameter is only supported for internal RASlog messages. Internal RASlog messages have an ID number equal to or greater than 5000.
Flag: read-only
Type: brocade-logging-types:ascii-text-type
Value: The message text corresponding to the internal RASlog message.
Config: false

current-severity

Description: The current severity level of the RASlog. The default severity 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.
Flag: read-write
Type: union (brocade-logging-types:severity-level-type and string)
Value: Valid values are info, warning, error, critical, and default.

default-severity

Description: The pre-defined default severity level of the RASlog.
Flag: read-only
Type: brocade-logging-types:severity-level-type
Value: Values displayed are info, warning, error, and critical.
Config: false

raslog-module

Description: A list of Fabric OS modules (identified by module ID) that have logging enabled.
Flag: read-write
Key: module-id
This list includes the following leafs:

module-id

Description: The Fabric OS module identified by module ID.
Flag: read-write
Type: brocade-logging-types:module-id-type
Value: A Fabric OS module ID.

log-enabled

Description: 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 (for example: GET https://10.10.10.10/rest/running/brocade-logging/raslog-module).
Flag: write-only

Type: boolean
Value: true = Logging is enabled for the Fabric OS module. **false** = Logging is disabled for the Fabric OS module.

log-quiet-control
Description: A list of parameters that control quiet time for logging based on the log type.
Key: log-type
Flag: read-write
This list includes the following leafs:

log-type
Description: The log type for which quiet time is configured.
Flag: read-write
Type: string
Value: audit = Quiet time is configured for audit logs. **raslog** = Quiet time is configured for RASlogs.

quiet-enabled
Description: Whether quiet time is enabled for logs.
Flag: read-write
Type: boolean
Value: true = Quiet time is enabled. **false** = Quiet time is disabled.

start-time
Description: The RAS quiet start time for the specified log type. This parameter is available only when quiet time is enabled (quiet-enabled = true).
Flag: read-write
Type: brocade-logging-types:time-24hr-type
Value: The quiet start time in the 24 hour clock format hh:mm (where hh is 00 to 23 and mm is 00 to 5).

end-time
Description: The RAS quiet end time for the specified log type. Start time must be configured before you can set the end time. This parameter is available only when quiet time is enabled (quiet-enabled = true).
Flag: read-write
Type: brocade-logging-types:time-24hr-type
Value: The quiet end time in the 24 hour clock format hh:mm (where hh is 00 to 23 and mm is 00 to 5).

days-of-week
Description: The days of the week for which you want to set quiet time for the specified log type.
Flag: read-write
This container has the following leaf:

day
Description: The day of the week for which you want to set quiet time for the specified log type. This parameter is available only when quiet time is enabled (quiet-enabled = true). If you set this to **everyday**, **mon**, **tue**, **wed**, **thu**, **fri**, **sat**, or **sun**, you must configure the start and end time.
Flag: read-write
Type: string
Value: 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**, **tue**, **wed**, **thu**, **fri**, **sat**, or **sun** = Enables quiet time for the specified days of the week for the specified start time and end time.

log-setting

Description: The system-wide parameters to control the logs.
Flag: read-write
This container has the following leaf:

syslog-facility-level

Description: 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.
Flag: read-write
Type: string
Value: Valid values are **log_local0**, **log_local1**, **log_local2**, **log_local3**, **log_local4**, **log_local5**, **log_local6**, and **log_local7** (default).

keep-alive-period

Description: The RASlog keep alive timeout.
Flag: read-write
Type: uint16
Value: 0 through **24** = Keeps logging alive for the specified time. **0** = Disables the keep alive timeout. Default = **1**.

Supported Methods

Only the OPTIONS, HEAD, GET, POST, PATCH, and DELETE operations are supported in this module.

History

Release version	History
Fabric OS 8.2.1	This API call was introduced.

brocade-logging Examples

This section provides examples for the brocade-logging module.

Viewing the Current Audit Log Configuration

The following example uses the GET request to determine the Access Gateway mode of the switch.

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|false>

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/<true|false>

URI Request

```
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 Request

```
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 Request

```
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 Request

```
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 PATCH request to configure a non-secure syslog server '17.14.15.16'.

Structure

PATCH *<base_URI>*/running/brocade-logging/syslog-server

URI Request

```
PATCH 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 “204 No Content” status appears in the headers.

Configuring a Secure Syslog Server

The following example uses the PATCH request to configure a secure syslog server '17.14.15.17'.

Structure

PATCH *<base_URI>*/running/brocade-logging/syslog-server

URI Request

```
PATCH 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 “204 No Content” 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 Request

```
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 Request

```
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 Request

```
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|false>

URI Request

```
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/<true|false>

URI Request

```
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|false>

URI Request

```
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/<true|false>

URI Request

```
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.

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 Request

```
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/message-id/<message_id>

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-logging/raslog/message-id/CONF-5000
```

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 Request

```
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|false>

URI Request

```
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/<true|false>

URI Request

```
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 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.

```
<?xml version="1.0"?>
<Response>
    <log-quiet-control>
        <log-type>audit</log-type>
        <quiet-enabled>false</quiet-enabled>
        <start-time/>
        <end-time/>
        <days-of-week/>
    </log-quiet-control>
    <log-quiet-control>
        <log-type>raslog</log-type>
        <quiet-enabled>false</quiet-enabled>
        <start-time/>
        <end-time/>
        <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|false>

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/<true|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/log-type/*log_type*

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-quiet-control/log-type/audit
```

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/log-type/*log_type*

URI Request

```
PATCH https://10.10.10.10/rest/running/brocade-logging/log-quiet-control/log-type/audit
```

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 Keep Alive 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.

brocade-maps

This module provides a detailed view of the Monitoring and Alerting Policy Suite features available in the Fabric OS REST API.

This topic 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*.

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?                ssp-state-type
       |  +--ro power-supply-health?          ssp-state-type
       |  +--ro fan-health?                   ssp-state-type
       |  +--ro wwn-health?                   ssp-state-type {maps-types:chassis-platform}?
       |  +--ro temperature-sensor-health?    ssp-state-type
       |  +--ro ha-health?                    ssp-state-type {maps-types:chassis-platform}?
       |  +--ro control-processor-health?     ssp-state-type {maps-types:chassis-platform}?
       |  +--ro core-blade-health?            ssp-state-type {maps-types:chassis-platform}?
       |  +--ro blade-health?                 ssp-state-type {maps-types:chassis-platform}?
       |  +--ro flash-health?                 ssp-state-type
       |  +--ro marginal-port-health?         ssp-state-type
       |  +--ro faulty-port-health?           ssp-state-type
       |  +--ro missing-sfp-health?           ssp-state-type
       |  +--ro error-port-health?            ssp-state-type
       |  +--ro expired-certificate-health?   ssp-state-type
       |  +--ro airflow-mismatch-health?      ssp-state-type
       +--ro system-resources
       |  +--ro cpu-usage?                    uint32
       |  +--ro memory-usage?                 uint32
       |  +--ro total-memory?                 uint32
       |  +--ro flash-usage?                  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-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?            inet:ip-address
|  +--rw test-email
|     +--rw subject?                  string
|     +--rw body?                     string
+--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
+--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
|  +--rw quiet-time?                  uint32
|  +--rw quiet-time-clear?            boolean
|  +--rw un-quarantine-timeout?       uint32
|  +--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
|  +--rw is-active-policy?            boolean
|  +--ro is-predefined-policy?        boolean
+--ro monitoring-system-matrix*  [monitoring-system group-type]
   +--ro monitoring-system?           maps-types:maps-monitoring-system-type
   +--ro dashboard-category?          maps-types:maps-dashboard-category-type
   +--ro group-type                   maps-types:maps-group-type-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
+--ro is-quiet-time-supported?      boolean
+--ro minimum-quiet-time?           boolean
+--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
```

URI Format

The URI format for this module takes the following form:

• *<base_URI>*/running/brocade-maps/switch-status-policy-report to display the Switch Status Policy report.

• *<base_URI>*/running/brocade-maps/group to display or configure groups.

• *<base_URI>*/running/brocade-maps/rule to display or configure rules.

• *<base_URI>*/running/brocade-maps/dashboard-rule to display the dashboard rule.

• *<base_URI>*/running/brocade-maps/maps-policy to display or configure MAPS policies.

• *<base_URI>*/running/brocade-maps/paused-cfg to display or configure the paused objects list.

• *<base_URI>*/running/brocade-maps/system-resources to display system resources (such as memory, CPU, and flash).

• *<base_URI>*/running/brocade-maps/monitoring-systems-matrix to display the monitoring system matrix.

• *<base_URI>*/running/brocade-maps/maps-config to display or configure display or modify the current configuration.

Parameters

brocade-maps

Description: The manageable MAPS features.
Flag: read-write
This container has the following leafs:

switch-status-policy-report

Description: The Switch Status Policy report. The SSP report provides the overall health status of the switch and includes enough information for you to investigate further, if necessary.
Flag: read-only
Config: false
This container has the following leafs:

switch-health

Description: The overall status of the switch. The switch state is determined by the state of one or more of its components (power supply, fan, wwn, and so on).
Flag: read-only
Type: ssp-state-type
Value:
unknown = The switch is in an unknown state.

down = The switch is in a down state and service is impacted. Review the health of the switch components to determine the cause of the down state. If a FRU is down, fix or replace it with a new FRU.
marginal = The switch is in a marginal state. You must take action immediately or service may be impacted. For example, if a FRU is marginal, fix or replace it with a new FRU. Review the health of the switch components to determine the cause of the marginal state.
healthy = The switch is within normal operating conditions.
Optional: Yes

power-supply-health
Description: The state of the power supplies.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The FRU is in an unknown state.
down = The FRU is in a down state and service is impacted. If a FRU is down, fix or replace it with a new FRU.
marginal = The FRU is in a marginal state. You must take action immediately or service may be impacted. If a FRU is marginal, fix or replace it with a new FRU.
healthy = The FRU is within normal operating conditions.
Optional: Yes

fan-health
Description: The state of the fans.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The FRU is in an unknown state.
down = The FRU is in a down state and service is impacted. If a FRU is down, fix or replace it with a new FRU.
marginal = The FRU is in a marginal state. You must take action immediately or service may be impacted. If a FRU is marginal, fix or replace it with a new FRU.
healthy = The FRU is within normal operating conditions.
Optional: Yes

wwn-health
Description: The state of the WWN cards.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The FRU is in an unknown state.
down = The FRU is in a down state and service is impacted. If a FRU is down, fix or replace it with a new FRU.
marginal = The FRU is in a marginal state. You must take action immediately or service may be impacted. If a FRU is marginal, fix or replace it with a new FRU.
healthy = The FRU is within normal operating conditions.
Optional: Yes

temperature-sensor-health
Description: The state of the temperature sensors.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The temperature sensor is in an unknown state.
down = The temperature sensor is in a down state and service is impacted.

marginal = The temperature sensor is in a marginal state. You must take action immediately or service may be impacted.
healthy = The temperature sensor is within normal operating conditions.
Optional: Yes

ha-health
Description: The state of the high availability (both control processors are in sync).
Flag: read-only
Type: ssp-state-type
Value:
unknown = The HA is in an unknown state.
down = The HA is in a down state and service is impacted.
marginal = The HA is in a marginal state. You must take action immediately or service may be impacted.
healthy = The HA is within normal operating conditions.
Optional: Yes

control-processor-health
Description: The state of the control processors.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The FRU is in an unknown state.
down = The FRU is in a down state and service is impacted. If a FRU is down, fix or replace it with a new FRU.
marginal = The FRU is in a marginal state. You must take action immediately or service may be impacted. If a FRU is marginal, fix or replace it with a new FRU.
healthy = The FRU is within normal operating conditions.
Optional: Yes

core-blade-health
Description: The state of the core blades.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The blade is in an unknown state.
down = The blade is in a down state and service is impacted. If a blade is down, fix or replace it with a new blade.
marginal = The blade is in a marginal state. You must take action immediately or service may be impacted. If a blade is marginal, fix or replace it with a new blade.
healthy = The blade is within normal operating conditions.
Optional: Yes

blade-health
Description: The state of the application blades.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The blade is in an unknown state.
down = The blade is in a down state and service is impacted. If a blade is down, fix or replace it with a new blade.
marginal = The blade is in a marginal state. You must take action immediately or service may be impacted. If a blade is marginal, fix or replace it with a new blade.
healthy = The blade is within normal operating conditions.
Optional: Yes

flash-health

Description: The state of the flash usage.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The flash usage is in an unknown state.
down = The flash usage is in a down state and service is impacted.
marginal = The flash usage is in a marginal state. You must take action immediately or service may be impacted.
healthy = The flash usage is within normal operating conditions.
Optional: Yes

marginal-port-health

Description: The state of the marginal ports
Flag: read-only
Type: ssp-state-type
Value:
unknown = The port is in an unknown state.
down = The port is in a down state and service is impacted.
marginal = The port is in a marginal state. You must take action immediately or service may be impacted.
healthy = The port is within normal operating conditions.
Optional: Yes

faulty-port-health

Description: The state of the faulty ports
Flag: read-only
Type: ssp-state-type
Value:
unknown = The port is in an unknown state.
down = The port is in a down state and service is impacted.
marginal = The port is in a marginal state. You must take action immediately or service may be impacted.
healthy = The port is within normal operating conditions.
Optional: Yes

missing-sfp-health

Description: The state of the SFP.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The SFP is in an unknown state.
down = The SFP is in a down state and service is impacted. If a SFP is down, fix or replace it with a new SFP.
marginal = The SFP is in a marginal state. You must take action immediately or service may be impacted. If a SFP is marginal, fix or replace it with a new SFP.
healthy = The SFP is within normal operating conditions.
Optional: Yes

error-port-health

Description: The state of the error ports.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The port is in an unknown state.

down = The port is in a down state and service is impacted.
marginal = The port is in a marginal state. You must take action immediately or service may be impacted.
healthy = The port is within normal operating conditions.
Optional: Yes

expired-certificate-health
Description: The state of the certificate.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The certificate is in an unknown state.
down = The certificate is in a down state and service is impacted.
marginal = The certificate is in a marginal state. You must take action immediately or service may be impacted.
healthy = The certificate is within normal operating conditions.
Optional: Yes

airflow-mismatch-health
Description: The state of the airflow. 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.
Flag: read-only
Type: ssp-state-type
Value:
unknown = The airflow is in an unknown state.
down = The airflow is in a down state and service is impacted.
marginal = The airflow is in a marginal state. You must take action immediately or service may be impacted.
healthy = The airflow is within normal operating conditions.
Optional: Yes

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.
Flag: read-only
Config: false
This container has the following leafs:

cpu-usage
Description: The percentage of CPU usage.
Flag: read-only
Type: uint32
Value: The percentage of CPU usage (0 to 100%).
Optional: Yes

memory-usage
Description: The percentage of memory usage.
Flag: read-only
Type: uint32
Value: The percentage of memory usage (0 to 100%).
Optional: Yes

total-memory
Description: The total memory usage in kilobytes.
Flag: read-only

Type: uint32
Value: The total memory usage in kilobytes.
Optional: Yes

flash-usage
Description: The percentage of flash usage.
Flag: read-only
Type: uint32
Value: The percentage of flash usage (0 to 100%).
Optional: Yes

paused-cfg
Description: A list of elements or element groups that you want to pause or resume monitoring. You can only pause or resume monitoring of ports, FCIP circuits, or SFPs.
Flag: read-write
Key: *group-type*
This list has the following leafs:

group-type
Description: The element group of which you want to pause or resume monitoring.
Flag: read-write
Type: enumeration
Value:
fc-port = The FC port group.
sfp = The FC SFP group. Note that the system may have different SFPs based on the speed or vendor.
circuit = The FCIP circuit group. An FCIP circuit is a virtual connection between two extension systems.
Optional: Yes

members
Description: A list of elements (ports, FCIP circuits, or SFPs). There must be at least one member in the list.
Flag: read-write
This list has the following leaf:

member
Description: An element (port, FCIP circuit, or SFP).
Flag: read-write
Type: string
Value: An element (port, FCIP circuit, or SFP).
Optional: Yes

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.
Flag: read-write
Key: *name*
This list has the following leafs:

name
Description: The group name. The group name must be unique; it is not case sensitive and can contain up to 32 characters.
Flag: read-write
Type: maps-types:maps-group-name-type
Value: 1 to 32 alphanumeric characters.
Optional: Yes

group-type

Description: The type of elements (such as, fc-port, sfp, fan, and so on) contained in the group.
Flag: read-write
Type: maps-types:maps-group-type-type
Value: power-supply = The power supply group type. The device may have multiple power supplies and may be integrated with a fan. **fan** = The fan group type. The device may have multiple fans may be integrated with power supplies. **fc-port** = The FC port group type. **sfp** = The FC SFP group type. The device may have different SFPs based on speed or vendor. **blade** = The blade group type. This group type manages all blades including core, CP, or switch blades. **circuit** = The FCIP circuit group type. **circuit-qos** = The circuit QOS traffic group type. Note that each circuit can carry multiple QOS traffic. **temperature-sensor** = The temperature sensor group type. **flash** = The flash memory group type. **switch** = The switch group type. **chassis** = The chassis group type. **cpu** = The CPU group type. **wwn** = The WWN card group type. **flow** = The flow group type. **tunnel** = The tunnel group type. **tunnel-qos** = The tunnel qos group type. **backend-port** = The back end group type. **ge-port** = The giga bit 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. **vtap-port** = The vtap port group type. **asic** = The ASIC group type.
Optional: Yes

group-feature

Description: The existing group feature name (such as node-wwn, port-name, or unknown).
Flag: read-write
Type: maps-group-feature-type
Value: node-wwn = The node WWN feature. **port-name** = The port name feature. **unknown** = The feature is unknown.
Optional: Yes

feature-pattern

Description: The feature pattern. Specifies the wildcard characters while defining the feature characteristics. The wildcard characters "*" for any string, "?" for any single character, "[expr]" for one character from the set specified in the expression, or '!' for negation of the string, are supported. If '!' is specified in the pattern, the pattern must be in single quotes. For example, if you create a group where group feature is "port-name" and the feature pattern is "brcdhost*", the group has a membership defined as ports with a port name that begins with "brcdhost."
Flag: read-write
Type: string
Value: The feature pattern
Optional: Yes

is-predefined

Description: 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.
Flag: read-only
Type: boolean
Config: false
Value: true = The group is system-defined. **false** = The group is user-defined.
Optional: Yes

is-modifiable

Description: Whether you can modify the group. You can modify all user-defined groups, and predefined port type groups, except for the ALL_QUARANTINED_PORTS group and any flow group (although the parameter is true).
Flag: read-only
Type: boolean
Config: false

Value: true = The system-defined group can be modified. **false** = The system-defined group cannot be modified.
Optional: Yes

members
Description: A list of members (groups) (such as, node-wwn or port-name). This parameter is available only when the group feature parameter is node WWN or port name (group-feature=node-wwn or port-name). There must be at least one member in the list.
Flag: read-write
This list has the following leaf:

member
Description: A member (group) (such as, node-wwn or port-name).
Flag: read-write
Type: string
Value: A group (such as, node-wwn or port-name).
Optional: Yes

maps-config
Description: The MAPS configuration for the switch. You can perform the following MAPS configurations using this container:
• View the current MAPS configuration.
• Define the actions to take on the switch when a threshold is triggered.
• Specify the e-mail addresses to which the alerts are sent.
• Delete all user-defined MAPS configurations related to rules, groups, policies, and so on.
Flag: read-write
This container has the following leafs:

actions
Description: The global MAPS actions list.
Flag: read-write
This container has the following leafs:

action
Description: The MAPS actions. You can enable one or more actions globally at the switch level or per rule.
Flag: read-write
Type: maps-types:maps-generic-action-type
Value: port-fence = This action immediately takes ports offline, which mightcause loss of traffic. **snmp-trap** = This action generates a message (called a “trap”) that notifies a management station when specific events occur on a switch. **raslog** = This action adds an entry to the switch event log for an individual switch. **sddq** = This action moves the traffic destined to a port affected by device-based latency to a low-priorityvirtual channel. This action does not disable the port, but it reduces the effect of its latency on other flows in the fabric. **un-quarantine** = This action releases the previously quarantined ports. **decommission** = This action takes a port offline without loss of traffic. **port-toggle** = This action temporarily disables a port and then re-enables it, allowing the port to reset and recover from some device-based issues. **e-mail** = This action sends information about the event to one or more specified e-mail addresses. The e-mail alert specifies thethreshold and describes the event, much like an error message. **fms** = This action MAPS sends a notification event information to the FICON management service. **vtap-uninstall** = This action uninstalls vTAP feature if the mirrored frame count exceeds 250K IOPS and encryption is enabled on the 16Gb/s-capable ASIC. If encryption is not enabled on the ASIC, vTAP is not uninstalled. **re-balance** = This action brings the port group state back to balance state. This may take 3 or more seconds. **sw-marginal** = This action places the switch into a marginal operating state. **sw-critical** = = This action places the switch into a down operating state. **sfp-marginal** = This action places the SFP into a marginal operating state.

Optional: Yes

decommission-cfg
Description: The decommission behavior (with-disable or impair). The default is with-disable.
Flag: read-write
Type: enumeration
Value: impair = The decommission behavior is to impair the link instead of a decommission and disable (or fence if the process fails). After this action triggers, the port remains online with no routes unless no other shortest path links exist. **with-disable** = The decommission behavior is to decommission the port with disable (or fence if the process fails).
Optional: Yes

recipient-address-list
Description: The recipient list for e-mail alerts.
Flag: read-write
This container contains the following leafs:

recipient-address
Description: The e-mail address (such as john@mmm.com or psingh@xyz.com) for recipients of e-mail alerts.
Flag: read-write
Type: string
Value: Up to 5 e-mail addresses for recipients of e-mail alerts. An e-mail address can be between 5 and 128 alphanumeric characters.
Optional: Yes

sender-address
Description: The e-mail address of the sender.
Flag: read-write
Type: string
Value: The e-mail address of the sender. An e-mail address can be between 5 and 128 alphanumeric characters.
Optional: Yes

domain-name
Description: The domain name. Enter none to clear the name.
Flag: read-write
Type: string
Value: The domain name.
Optional: Yes

relay-ip-address
Description: The relay IP address. Enter none to clear the IP address.
Flag: read-write
Type: inet:ip-address
Value: The relay IP address.
Optional: Yes

test-email
Description: The test e-mail container.
Flag: read-write
This container has the following leafs:

subject
Description: The subject for the e-mail alert.
Flag: read-write
Type: string

Value: 0 to 256 alpha characters.
Optional: Yes

body

Description: The message for the e-mail alert.
Flag: read-write
Type: string
Value: 0 to 512 alpha characters.
Optional: Yes

dashboard-rule

Description: A list of dashboards. The dashboard enables you to view the events or rules 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.
Flag: read-only
Config: false
This container has the following leafs:

category

Description: The dashboard category. Each rule can belong to only one category.
Flag: read-only
Type: maps-types:maps-dashboard-category-type
Value: 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 FRU Health category enables you to define rules for field replacable units (FRUs). **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 monitors 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 replacable 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 monitors your system's temperature, flash usage, memory usage, and CPU. **fabric-performance-impact** = The Fabric Performance Impact category monitors the current condition of the latency detected on E_Ports and F_Ports in a fabric over different time windows and uses this to determine the performance impact to the fabric and network **traffic-performance** = The Traffic Performance category monitors flows.
Optional: Yes

name

Description: The rule name.
Flag: read-only
Type: maps-types:maps-rule-name-type
Value: 1 to 72 alphanumeric character rule name.
Optional: Yes

triggered-count

Description: The number of times the rule was triggered for the category.

Flag: read-only
Type: uint32
Value: The number of times the rule was triggered for the category.
Optional: Yes

time-stamp
Description: The date and time that the rule was last triggered.
Flag: read-only
Type: yang:date-and-time
Value: The date and time that the rule was last triggered.
Optional: Yes

repetition-count
Description: 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.
Flag: read-only
Type: uint32
Value: The number of times a rule was triggered.
Optional: Yes

objects
Description: The objects that violated the rule. For example, port, circuit, and so on.
Flag: read-only
This container has the following leaf:

object
Description: The object that violated the rule. For example, port, circuit, and so on. The object format is as follows: <element>:<value>. For example, 'F-Port 10:90' and 'U-Port 11:11'.
Flag: read-only
Type: string
Value: The object that violated the rule.
Optional: Yes

dashboard-misc
Description: The dashboard miscellaeous infomation (such as start time and operation).
Flag: read-write
This container has the following leaf:

maps-start-time
Description: The MAPS start time. MAPS is restartable service which means the start time could be different than system up time.
Flag: read-only
Type: yang:date-and-time
Config: false
Value: The MAPS start time.
Optional: Yes

clear-data
Description: Whether to clear the dashboard data.
Flag: read-write
Type: boolean
Value: true = Clears the dashboard data. **false** = Does not clear the dashboard data.
Optional: Yes

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.
Flag: read-write
Key: *name*
This list has the following leafs:

name

Description: The rule name.
Flag: read-write
Type: maps-types:maps-rule-name-type
Value: 1 to 72 alphanumeric character rule name.
Optional: Yes

is-rule-on-rule

Description: 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.
Flag: read-write
Type: maps-types:maps-rule-type
Value: true = It is rule-on-rule. **false** = It is a base rule.
Mandatory: true
Optional: Yes

monitoring-system

Description: The statistic or error to be monitored (such as CRC, ITW, PS_STATE, and so on).
Flag: read-write
Type: maps-types:maps-monitoring-system-type
Value: The statistic or error to be monitored (such as CRC, ITW, PS_STATE, and so on)
Mandatory: true
Optional: Yes

time-base

Description: The time interval between two samples to be compared.
Flag: read-write
Type: maps-types:maps-time-base-type
Value: none = The time base is not applicable. **second** = The samples are compared every second. **minute** = The samples are compared every minute. **five-minute** = The samples are compared every five minutes. **hour** = The samples are compared every hour. **day** = The samples are compared every day. **week** = The samples are compared every week.
Mandatory: true
Optional: Yes

logical-operator

Description: The relational operation to be used in evaluating the condition.
Flag: read-write
Type: maps-types:maps-logical-operator-type
Value: 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.
Mandatory: true
Optional: Yes

threshold-value

Description: 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.
• For numerical values: 0-999999999. The upper limit may vary depending on the monitoring system category.
• For percentage values: 0-100.
• For FRU states: ON, OFF, IN, OUT, or FAULTY.
• For temperature monitoring: IN_RANGE or OUT_OF_RANGE.
• For FPI states: IO_FRAME_LOSS, IO_PERF_IMPACT, IO_LATENCY_CLEAR.
• For Ethernet port state: UP or DOWN
Flag: read-write
Type: maps-types:threshold-value-type
Value: The threshold value.
Mandatory: true
Optional: Yes

group-name

Description: The group name. The group name must be unique; it is not case sensitive and can contain up to 32 characters.
Flag: read-write
Type: maps-types:maps-group-name-type
Value: The group name.
Mandatory: true
Optional: Yes

actions

Description: The MAPS actions.
Flag: read-write
This container has the following leaf:

action

Description: The MAPS action.
Flag: read-write
Type: maps-types:maps-generic-action-type
Value:
Value: port-fence = This action immediately takes ports offline, which mightcause loss of traffic. **snmp-trap** = This action generates a message (called a “trap”) that notifies a management station when specific events occur on a switch. **raslog** = This action adds an entry to the switch event log for an individual switch. **sddq** = This action moves the traffic destined to a port affected by device-based latency to a low-priorityvirtual channel. This action does not disable the port, but it reduces the effect of its latency on other flows in the fabric. **un-quarantine** = This action releases the previously quarantined ports. **decomission** = This action takes a port offline without loss of traffic. **port-toggle** = This action temporarily disables a port and then re-enables it, allowing the port to reset and recover from some device basedissues. **e-mail** = This action sends information about the event to one or more specified e-mail addresses. The e-mail alert specifies thethreshold and describes the event, much like an error message. **f's** = This action MAPS sends a notification event information to the FICON management service. **vtap-uninstall** = This action uninstalls vTAP feature if the mirrored frame count exceeds 250K IOPS and encryption is enabled on the 16Gb/s-capable ASIC. If encryption is not enabled on the ASIC, vTAP is not uninstalled. **re-balance** = This action brings the port group state back to balance state. This may take 3 or more seconds. **sw-marginal** = This action places the switch into a marginal operating state. **sw-critical** = = This action places the switch into a down operating state. **sfp-marginal** = This action places the SFP into a marginal operating state.
Optional: Yes

is-predefined
Description: 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.
Flag: read-only
Type: boolean
Config: false
Value: true = The group is system-defined. **false** = The group is user-defined.
Optional: Yes

event-severity
Description: The user-configured severity (such as warning, error, critical, info, or default).
Flag: read-write
Type: maps-types:maps-event-severity-type
Value: The user-configured severity (such as warning, error, critical, info, or default).
Optional: Yes

toggle-time
Description: The port's toggle time in seconds.
Flag: read-write
Type: uint32
Value: 2 to 3600 seconds.
Optional: Yes

quiet-time
Description: The quiet time in seconds. The rule is not triggered until quiet time has expired.
Flag: read-write
Type: uint32
Value: 60 to 31536000 seconds.
Optional: Yes

quiet-time-clear
Description: Whether to clear quiet time.
Flag: read-write
Type: boolean
Value: true = Clears the quiet time from the rule. **false** = Does not clear the quiet time from the rule.
Optional: Yes

un-quarantine-timeout
Description: The unquarantine timeout in seconds.
Flag: read-write
Type: uint32
Value: 0 to 2147483647 seconds.
Optional: Yes

un-quarantine-clear
Description: Whether to clear the unquarantine timeout.
Flag: read-write
Type: boolean
Value: true = Clears the unquarantine timeout from the rule. **false** = Does not clear the unquarantine timeout from the rule.
Optional: Yes

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.
Flag: read-write

This container has the following leafs:

name

Description: The MAPS policy name. The name for the policy must be unique; it is case-sensitive and can contain up to 32 characters.
Flag: read-write
Type: maps-types:maps-policy-name-type
Value: The MAPS policy name. 1 to 32 alphanumeric characters and underscores.
Optional: Yes

rule-list

Description: A list of rules in the policy
Flag: read-write
This container has the following leaf:

rule

Description: The rule name.
Flag: read-write
Type: maps-types:maps-rule-name-type
Value: The MAPS rule name. 1 to 72 alphanumeric characters and underscores.
Optional: Yes

is-active-policy

Description: Whether the policy is active. You can configure multiple policies; however, only one policy can be active at a time.
Flag: read-write
Type: boolean
Value: true = This policy is active. **false** = This policy is not active.
Optional: Yes

is-predefined-policy

Description: 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.
Flag: read-write
Type: boolean
Value: true = This policy is predefined. **false** = This policy is user-defined.
Optional: Yes

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.
Flag: read-only
Key: monitoring-system group-type
This list has the following leafs:

monitoring-system

Description: The monitoring system name (CRC, BLADE_STATE, ITW, and so on). A 'monitoring system' is a value (measure or statistic) that can be monitored. For information about MAPS monitoring systems, refer to *MAPS monitoring categories* in the *Brocade Monitoring and Alerting Policy Suite Configuration Guide*.
Flag: read-only
Type: maps-types:maps-monitoring-system-type
Value: The monitoring system name (CRC, BLADE_STATE, ITW, and so on).
Optional: Yes

dashboard-category

Description: The dashboard category of the monitoring system.
Flag: read-only
Type: maps-types:maps-dashboard-category-type
Value:
Value: port-health = The Port Health category monitors port statistics. **backend-port-health** = The FRU Health category monitors the health of field replacable units (FRUs). **extension-ge-port-health** = The Gigabit Ethernet (GE) ports category monitors statistics for GE ports. **security-violations** = The Security Health category monitors security violations on the switch. **fabric-state-changes** = The Fabric State Changes category monitors areas of potential fabric related or switch related problems. **FRU-health** = The FRU Health category monitors field replacable units (FRUs). **extension-health** = The Extension Health category monitors Extension health. **switch-resources** = The Switch Resource category monitors your system's temperature, flash usage, memory usage, and CPU. **fabric-performance-impact** = The Fabric Performance Impact category monitors the current condition of the latency detected on E_Ports and F_Ports in a fabric over different time windows and uses this to determine the performance impact to the fabric and network **traffic-performance** = The Traffic Performance category monitors flows.
Optional: Yes

group-type

Description: The group type.
Flag: read-only
Type: maps-types:maps-group-type-type
Value: power-supply = The power supply group type. The device may have multiple power supplies and may be integrated with a fan. **fan** = The fan group type. The device may have multiple fans may be integrated with power supplies. **fc-port** = The FC port group type. **sfp** = The FC SFP group type. The device may have different SFPs based on speed or vendor. **blade** = The blade group type. This group type manages all blades including core, CP, or switch blades. **circuit** = The FCIP circuit group type. **circuit-qos** = The circuit QOS traffic group type. Note that each circuit can carry multiple QOS traffic. **temperature-sensor** = The temperature sensor group type. **flash** = The flash memory group type. **switch** = The switch group type. **chassis** = The chassis group type. **cpu** = The CPU group type. **wwn** = The WWN card group type. **flow** = The flow group type. **tunnel** = The tunnel group type. **tunnel-qos** = The tunnel qos group type. **backend-port** = The back end group type. **ge-port** = The giga bit 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. **vtap-port** = The vtap port group type. **asic** = The ASIC group type.
Optional: Yes

base-time-bases

Description: The time bases.
Flag: read-only
This container has the following leaf:

time-base

Description: A list of the supported time bases for the monitoring system.
Flag: read-only
Type: maps-types:maps-time-base-type
Value: none = The time base is not applicable. **second** = The samples are compared every second. **minute** = The samples are compared every minute. **five-minute** = The samples are compared every five minutes. **hour** = The samples are compared every hour. **day** = The samples are compared every day. **week** = The samples are compared every week.
Optional: Yes

rule-on-rule-time-bases

Description: The rule on rules time bases for the monitoring system. This parameter is available only when the is-rule-on-rule-supported parameter is active (is-rule-on-rule-supported =true)

Flag: read-only
This container has the following leaf:

rule-on-rule-time-base
Description: A list of the supported time bases for rule on ules for the monitoring system.
Flag: read-only
Type: maps-types:maps-time-base-type
Value: none = The time base is not applicable. **second** = The samples are compared every second. **minute** = The samples are compared every minute. **five-minute** = The samples are compared every five minutes. **hour** = The samples are compared every hour. **day** = The samples are compared every day. **week** = The samples are compared every week.
Optional: Yes

is-read-only
Description: Whether the monitoring system is read only.
Flag: read-only
Type: boolean
Value: true = The monitoring system is read only. **false** = The monitoring system is not read only.
Optional: Yes

monitored-logical-switch
Description: Whether the monitoring system is in all logical switches or only the default logical switch.
Flag: read-only
Type: enumeration
Value: all-logical-switches = The monitoring system is in all logical switches. **default-switch-only** = The monitoring system is only in the default logical switch.
Optional: Yes

is-rule-on-rule-supported
Description: Whether rule on rule is supported.
Flag: read-only
Type: boolean
Value: true = Rule on rule is supported. **false** = Rule on rule is not supported.
Optional: Yes

is-quiet-time-supported
Description: Whether quiet time is supported.
Flag: read-only
Type: boolean
Value: true = Quiet time is supported. **false** = Quiet time is not supported.
Optional: Yes

minimum-quiet-time
Description: The minimum quiet time in seconds.
Flag: read-only
Type: uint32
Value: 60 to 31536000 seconds.
Optional: Yes

monitoring-type
Description: The monitoring type (event based or poll based) for the monitoring system.
Flag: read-only
Type: maps-types:monitoring-type-type
Value: event-based = Event based monitoring. **poll-based** = Poll based monitoring.
Optional: Yes

data-type

Description: The data type support. Each monitoring system supports different data types for thresholds.
Flag: read-only
Type: maps-types:data-type-type
Value: unsigned-int32 = The unsigned integer 32 bits data type. **int32** = The integer 32 bits data type. **float** = The float data type **unsigned-int64** = The unsigned int 64 bits data type. **enum** = The enums data type.
Optional: Yes

description

Description: A description of the monitoring system.
Flag: read-only
Type: string
Value: A 1 to 128 character description of the monitoring system.
Optional: Yes

actions

Description: The global MAPS actions list.
Flag: read-only
This container has the following leafs:

action

Description: A list of the MAPS actions defined for this monitoring system.
Flag: read-only
Type: maps-types:maps-generic-action-type
Value: port-fence = The port fence action immediately takes ports offline, which mightcause loss of traffic. **snmp-trap** = This action generates a message (called a “trap”) that notifies a management station when specific events occur on a switch. **raslog** = This action adds an entry to the switch event log for an individual switch. **sddq** = This action moves the traffic destined to a port affected by device-based latency to a low-priorityvirtual channel. This action does not disable the port, but it reduces the effect of its latency on other flows in the fabric. **un-quarantine** = This action releases the previously quarantined ports. **decomission** = This action takes a port offline without loss of traffic. **port-toggle** = This action temporarily disables a port and then re-enables it, allowing the port to reset and recover from some device basedissues. **e-mail** = This action sends information about the event to one or more specified e-mail addresses. The e-mail alert specifies thethreshold and describes the event, much like an error message. **f's** = This action MAPS sends a notification event information to the FICON management service. **vtap-uninstall** = This action uninstalls vTAP feature if the mirrored frame count exceeds 250K IOPS and encryption is enabled on the 16Gb/s-capable ASIC. If encryption is not enabled on the ASIC, vTAP is not uninstalled. **re-balance** = This action brings the port group state back to balance state. This may take 3 or more seconds. **sw-marginal** = This action places the switch into a marginal operating state. **sw-critical** = = This action places the switch into a down operating state. **sfp-marginal** = This action places the SFP into a marginal operating state.
Optional: Yes

unit

Description: The data unit of the monitoring system.
Flag: read-only
Type: maps-types:maps-data-unit
Value: CRCs, **ITWs**, **timeouts**, **milli-ampere**, **days**, **errors**, **loss-of-signals**, **loss-of-synchronizations**, **violations**, **ports**, **extension-flows**, **it-flows**, **certificates**, **segmentations**, **changes**, **logins**, **IOs**, **IOPS**, **MBps**, **%**, **hours**, **centigrade**, **bytes**, **micro-seconds**, **milli-seconds**, **milli-volts**, **milli-amps**, and **micro-watts**
Optional: Yes

data-range

Description: The data range of the monitoring system.
Flag: read-only
Type: string
Value: 1 to 100 alphanumeric characters plus hyphens, commas, and underscore characters.
Optional: Yes

logical-operators

Description: The supported operations.
Flag: read-only
This container has the following leaf:

logical-operator

Description: The relational operation to be used in evaluating the condition.
Flag: read-only
Type: maps-types:maps-logical-operator-type
Value: 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.
Optional: Yes

Supported Methods

Only the OPTIONS, HEAD. GET, DELETE, PATCH, and POST operations are supported in this module.

History

Release version	History
Fabric OS 8.2.1	This API call was introduced.

brocade-maps Examples

This section provides examples for the brocade-maps module.

This section 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>
    </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 port 8/9, 8/17, 8/18 and 6/23 in the f-ports group.

Structure

POST *<base_URI>*/running/brocade-maps/paused-cfg/group-type/fc-port

URI

```
POST https://10.10.10.10/rest/running/brocade-maps/paused-cfg/group-type/fc-port
```

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 port 8/17, 8/18 and 6/23 in the f-ports group by deleting them from the paused configuration.

Structure

DELETE *<base_URI>*/running/brocade-maps/paused-cfg/group-type/fc-port

URI

```
DELETE https://10.10.10.10/rest/running/brocade-maps/paused-cfg/group-type/fc-port
```

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 “201 Created” 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 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>8/5</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 an existing group ALL_FCOE_100G_SR4_QSFP with two ports 5/8 and 5/17.

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_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 “201 Created” status message.

Viewing the MAPS Configuration

The following example uses the GET request to display the current MAPS configuration.

Structure

GET *<base_URI>*/running/brocade-maps/maps-config

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/maps-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>
    <maps-config>
        <recipient-address-list>
            <recipient-address>john.doe@company.com</recipient-address>
        </recipient-address-list>
        <relay-ip-address/>
        <domain-name/>
        <actions>
            <action>raslog</action>
            <action>e-mail</action>
            <action>sw-critical</action>
            <action>sw-marginal</action>
        </actions>
        <sender-address>john.doe@company.com</sender-address>
        <decommission-cfg>impair</decommission-cfg>
    </maps-config>
</Response>
```

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'.

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>
```

```
        <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>
    </maps-config>
```

Response Body

When the operation is successful, the response has an empty message body and a “200 OK” status message.

Viewing the MAPS Dashboard

The following example uses the GET request to display the MAPS dashboard.

Structure

GET *<base_URI>*/running/brocade-maps/dashboard-rule

URI

```
GET https://10.10.10.10/rest/running/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 appears in the headers.

```
<?xml version="1.0"?>
<Response>
    <dashboard-rule>
        <category>Fabric State Changes</category>
        <triggered-count>1</triggered-count>
        <name>defSWITCHEPORT_DOWN_1</name>
        <time-stamp>Jun 28 11:26:13</time-stamp>
        <repetition-count>1</repetition-count>
        <objects>
            <object>Switch:2</object>
        </objects>
    </dashboard-rule>
</Response>
```

Viewing MAPS Dashboard Settings

The following example uses the GET request to display the date and time that 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 following provides 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 a 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-clear/true

URI

```
PATCH https://10.10.10.10/rest/running/brocade-maps/rule/name/BaseRule_statechanges12/quiet-time-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.

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.

Viewing MAPS Policies

The following example uses the GET request to display the MAPS policy configurations. To display a specific MAPS policy, specify the policy name in the request.

For example, the following provides specifying the rule name URI structure and URI:

GET *<base_URI>*/running/brocade-maps/maps-policy/name/*policy_name*.

```
GET https://10.10.10.10/rest/running/brocade-maps/maps-policy/name/defALL_100G_QSFPCURRENT_10
```

Structure

GET *<base_URI>*/running/brocade-maps/maps-policy

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/maps-policy
```

Request Body

No request body is required.

```
<?xml version="1.0"?>
<Response>
    <maps-policy>
        <name>dflt_aggressive_policy</name>
        <rule-list>
            <rule>defALL_100G_QSFPCURRENT_10</rule>
            <rule>defALL_100G_QSFPCURRENT_2</rule>
            <rule>defALL_100G_QSFPRXP_2187</rule>
            <rule>defALL_100G_QSFPRXP_60</rule>
            <rule>defALL_100G_QSFPSFP_TEMP_75</rule>
            <rule>defALL_100G_QSFPSFP_TEMP_n5</rule>
            <rule>defALL_100G_QSFPTXP_3467</rule>
            <rule>defALL_100G_QSFPTXP_48</rule>
            <rule>defALL_100G_QSFPVOLTAGE_2970</rule>
            <rule>defALL_100G_QSFPVOLTAGE_3630</rule>
.
.
.
            <rule>defSWITCHSEC_TS_D2</rule>
            <rule>defSWITCHSEC_TS_H1</rule>
            <rule>defSWITCHZONE_CFGSZ_PER_70</rule>
            <rule>defSWITCHZONE_CHG_2</rule>
        </rule-list>
        <is-predefined-policy>false</is-predefined-policy>
        <is-active-policy>false</is-active-policy>
```

```
    </maps-policy>
</Response>
```

Response Body

When the operation is successful, the response has an empty message body and a “200 OK” status message.

Configuring a MAPS Policy

The following example uses the POST request to create a 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 a 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 MAPS Monitoring System Matrix

The following example uses the GET request to display the MAPS monitoring system matrix.

Structure

GET *<base_URI>*/running/brocade-maps/monitoring-system-matrix

URI

```
GET https://10.10.10.10/rest/running/brocade-maps/monitoring-system-matrix
```

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>
    <monitoring-system-matrix>
        <monitoring-system>CRC</monitoring-system>
        <data-type>unsigned-int32</data-type>
        <data-range>0-999999999</data-range>
        <group-type>fc-port</group-type>
        <dashboard-category>Port</dashboard-category>
        <description>CRC errors (CRC)</description>
        <unit>CRCs</unit>
        <base-time-bases>
            <time-base>min</time-base>
            <time-base>hour</time-base>
            <time-base>day</time-base>
        </base-time-bases>
        <actions>
            <action>raslog</action>
            <action>snmp-trap</action>
            <action>e-mail</action>
            <action>port-fence</action>
            <action>decomission</action>
            <action>fms</action>
        </actions>
        <is-quiet-time-supported>true</is-quiet-time-supported>
        <is-rule-on-rule-supported>true</is-rule-on-rule-supported>
        <rule-on-rule-time-bases>
            <rule-on-rule-time-base>hour</rule-on-rule-time-base>
            <rule-on-rule-time-base>day</rule-on-rule-time-base>
            <rule-on-rule-time-base>week</rule-on-rule-time-base>
        </rule-on-rule-time-bases>
        <monitoring-type>poll-based</monitoring-type>
        <monitored-logical-switch>all-logical-switches</monitored-logical-switch>
        <logical-operators>
            <logical-operator>g</logical-operator>
            <logical-operator>ge</logical-operator>
        </logical-operators>
        <minimum-quiet-time>60</minimum-quiet-time>
```

```
            <is-read-only>false</is-read-only>
        </monitoring-system-matrix>
    .
    .
    .
        <monitoring-system-matrix>
            <monitoring-system>DEV_LOGIN_DIST</monitoring-system>
            <data-type>enum</data-type>
            <data-range>BALANCED,IMBALANCED,BALANCE_FAILED</data-range>
            <group-type>fc-port</group-type>
            <dashboard-category>Fabric Performance</dashboard-category>
            <description>Device logins distribution (DEV_LOGIN_DIST)</description>
            <unit/>
            <base-time-bases>
                <time-base>none</time-base>
            </base-time-bases>
            <actions>
                <action>raslog</action>
                <action>snmp-trap</action>
                <action>e-mail</action>
                <action>port-fence</action>
                <action>decomission</action>
                <action>fms</action>
                <action>re-balance</action>
            </actions>
            <is-quiet-time-supported>true</is-quiet-time-supported>
            <is-rule-on-rule-supported>true</is-rule-on-rule-supported>
            <rule-on-rule-time-bases>
                <rule-on-rule-time-base>hour</rule-on-rule-time-base>
                <rule-on-rule-time-base>day</rule-on-rule-time-base>
                <rule-on-rule-time-base>week</rule-on-rule-time-base>
            </rule-on-rule-time-bases>
            <monitoring-type>event-based</monitoring-type>
            <monitored-logical-switch>all-logical-switches</monitored-logical-switch>
            <logical-operators>
                <logical-operator>eq</logical-operator>
            </logical-operators>
            <minimum-quiet-time>60</minimum-quiet-time>
            <is-read-only>false</is-read-only>
        </monitoring-system-matrix>
    </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..

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 Types

```
module: brocade-media
  +--ro brocade-media
     +--ro media-rdp* [name]
        +--ro name                                  interface-type
        +--ro identifier?                           string
        +--ro connector?                            string
        +--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 peer-data-available?                  boolean
        +--ro remote-identifier?                    string
        +--ro remote-laser-type?                    string
        +--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
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
```

for data type descriptions.

URI format

The URI format for this module takes one of the following forms:

• *<base_URI>*/running/brocade-media/media-rdp followed by the leafs as listed in the module tree to display a summary of the SFP media data for all available ports.

• *<base_URI>*/running/brocade-media/media-rdp/name/ followed by the leafs as listed in the module tree to display media data for a specific SFP.

Supported methods

Only the OPTIONS, GET, and HEAD operations are supported in this module.

History

Release version	History
Fabric OS 8.2.1	This API call was introduced.
Fabric OS 9.0.0	This API call was modified to add the peer-data-available parameter.
Fabric OS 9.0.1	This API call was modified to include support for the 64-Gb/s SFPs for the power-on-time parameter.
Fabric OS 9.1.0	This API call was modified to add the remote-media-voltage-alarm-type, container remote-media-temperature-alarm-type, remote-media-tx-bias-alarm-type, remote-media-tx-power-alarm-type, and remote-media-rx-power-alarm-type containers and the alarm-type and temp-alarm-type definitions. This API call was modified to edit the identifier and connector leafs, the date-code-type definition, and the media-optical-product-data-group. This API call was modified to make obsolete the remote-media-voltage-alert (use remote-media-voltage-alarm-type), remote-media-temperature-alert (use remote-media-temperature-alarm-type), remote-media-tx-bias-alert (use remote-media-tx-bias-alarm-type), remote-media-tx-power-alert (use remote-media-tx-power-alarm-type), and remote-media-rx-power-alert (use remote-media-rx-power-alarm-type) containers, the high-alarm, low-alarm, high-warning, and low-warning leafs, and the alert-type definition (use alarm-type).

brocade-name-server

This module is used to monitor the operation of one or more instances of Name Server functionality.

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
```

URI Format

The URI format for this module takes one of the following forms:

• *<base_URI>*/running/brocade-name-server/fibrechannel-name-server to view Name Server attributes for the switch.

• *<base_URI>*/running/brocade-name-server/fibrechannel-name-server/port-id to view the Name Server attributes for a specific port.

Supported Methods

Only the GET, HEAD, and OPTIONS operations are supported in this module.

History

Release version	History
Fabric OS 8.2.0a	This API call was introduced.
Fabric OS 8.2.1	Refined descriptions and headings.
Fabric OS 9.0.0	This API call was modified to edit the lsan leaf and to make the share-area leaf is obsolete.
Fabric OS 9.1.0	This API call was modified to add the protocol-speed and connected-through-fc-lag leafs. This API call was modified to edit the port-id, port-name, and node-name leafs and the fibrechannel-name-server list. This API call was modified to deprecate the link-speed leaf (use the protocol-speed leaf).

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*.

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
       |  +--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 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-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
|  +--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
+--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                      brocade-security-type:user-config-role-type
|  +--rw switch-role?                   brocade-security-type:user-config-role-type
|  +--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?   brocade-security-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
+--ro sec-crypto-cfg
|  +--ro ssh-cipher?                    brocade-security-type:default-string-type
|  +--ro ssh-kex?                       brocade-security-type:default-string-type
|  +--ro ssh-mac?                       brocade-security-type:default-string-type
|  +--ro https-cipher?                  brocade-security-type:default-string-type
|  +--ro radius-cipher?                 brocade-security-type:default-string-type
|  +--ro ldap-cipher?                   brocade-security-type:default-string-type
|  +--ro syslog-cipher?                 brocade-security-type:default-string-type
|  +--ro https-tls-protocol?            brocade-security-type:tls-protocol-type
|  +--ro radius-tls-protocol?           brocade-security-type:tls-protocol-type
|  +--ro ldap-tls-protocol?             brocade-security-type:tls-protocol-type
|  +--ro syslog-tls-protocol?           brocade-security-type:tls-protocol-type
|  +--ro x509v3-validation-mode?        brocade-security-type:validation-mode-type
+--rw sshutil
|  +--rw allow-user-name?               brocade-security-type:user-config-user-name-type
|  +--rw rekey-interval?                uint16
+--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
|  +--rw passphrase?                    string
|  +--ro size?                          uint16
|  +--ro fingerprint?                   string
+--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 country-name?                string
|  +--rw state-name?                  string
|  +--rw locality-name?               string
|  +--rw organization-name?           string
|  +--rw unit-name?                   string
|  +--rw domain-name?                 inet:host
+--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 operation?                   brocade-security-type:seccertmgmt-operation-type
+--ro security-certificate* [certificate-entity certificate-type]
   +--ro certificate-entity           brocade-security-type:certificate-entity-type
   +--ro certificate-type             brocade-security-type:certificate-application-type
   +--ro certificate?                 string
   +--ro certificate-hexdump?         string
```

URI Format

The URI format for this module takes the following form:

- *<base_URI>*/running/brocade-security/ipfilter-policy followed by the leafs as listed in the module tree to display or configure IP filter policies.

• *<base_URI>*/running/brocade-security/ipfilter-rule followed by the leafs as listed in the module tree to display or configure IP filter rules.

• *<base_URI>*/running/brocade-security/user-specific-password-cfg to display or configure user-specific password policy configurations.

• *<base_URI>*/running/brocade-security/password-cfg followed by the leafs as listed in the module tree to display or configure password policy configurations.

• *<base_URI>*/running/brocade-security/user-config followed by the leafs as listed in the module tree to display or configure user configurations.

• *<base_URI>*/running/brocade-security/auth-spec followed by the leafs as listed in the module tree to display or configure authentication modes.

• *<base_URI>*/running/brocade-security/radius-server followed by the leafs as listed in the module tree to display or configure a RADIUS server.

• *<base_URI>*/running/brocade-security/tacacs-server followed by the leafs as listed in the module tree to display or configure a TACACS+ server.

• *<base_URI>*/running/brocade-security/ldap-server followed by the leafs as listed in the module tree to display or configure a LDAP server.

• *<base_URI>*/running/brocade-security/ldap-role-map followed by the leafs as listed in the module tree to display or configure LDAP role mappings.

• *<base_URI>*/running/brocade-security/sec-crypto-cfg-template-action followed by the leafs as listed in the module tree to activate, verify, import, or export SecCryptoCfg templates.

• *<base_URI>*/running/brocade-security/sec-crypto-cfg-template followed by the leafs as listed in the module tree to display the SecCryptoCfg templates.

• *<base_URI>*/running/brocade-security/sec-crypto-cfg followed by the leafs as listed in the module tree to display the active Security Crypto Configuration.

• *<base_URI>*/running/brocade-security/sshutil followed by the leafs as listed in the module tree to display or configure the allowed user and rekey configuration.

• *<base_URI>*/running/brocade-security/sshutil-key followed by the leafs as listed in the module tree to display or generate a Host key, or delete an SSH Host and pub-private keys.

• *<base_URI>*/running/brocade-security/sshutil-public-key followed by the leafs as listed in the module tree to display or configure public keys for a specific user or to delete the public keys for a specified user or all users.

• *<base_URI>*/running/brocade-security/sshutil-public-key-action followed by the leafs as listed in the module tree to import or export a public key to or from a remote host.

• *<base_URI>*/running/brocade-security/password followed by the leafs as listed in the module tree to configure a user's password .

• *<base_URI>*/running/brocade-security/security-certificate-generate followed by the leafs as listed in the module tree to generate a third-party certificate on a device.

• *<base_URI>*/running/brocade-security/security-certificate-action followed by the leafs as listed in the module tree to import or export a security certificate to or from a remote host.

• *<base_URI>*/running/brocade-security/security-certificate followed by the leafs as listed in the module tree to display a security certificate.

Parameters

brocade-security

Description: The container for security configuration.
Flag: read-write
This container has the following containers and lists:

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 filter policies can either be in the defined configuration or in the active configuration. There can be a 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 IP policies in the defined configuration. Note that only active policies are enforced.
Flag: read-write
Key: name
This list has the following leafs:

name

Description: The IP filter policy name.
Flag: read-write
Type: brocade-security-type:ipfilter-name-type
Value: 1 to 20 alphanumeric and underscore characters. The policy name must be unique. Note that policy names are case-insensitive and are always stored as lower case. The IP filter policy names 'default_ipv4' and 'default_ipv6' are reserved for the default IP filter policies.
Optional: No

ip-version

Description: The IP filter policy version type (such as IPv4 or IPv6).
Flag: read-write
Type: brocade-security-type:ipfilter-ip-version-type
Value: IPv4 = This creates an IPv4 policy. **IPv6** = This creates an IPv6 policy.
Optional: Yes

is-policy-active

Description: The current status of the policy (defined or active).
Flag: read-only
Type: boolean
Config: false
Value: true = The active policy. **False** = The defined policy.
Optional: Yes

is-default-policy

Description: Whether the policy is a default or user-defined policy.
Flag: read-only
Type: boolean
Config: false
Value: true = The default policy. **False** = The user-defined policy.
Optional: Yes

action

Description: The action to take on the IP filter policy. IP filter configurations are saved automatically after any change. You can set this parameter with a PATCH request; however, a GET request does not fetch this parameter.
Flag: write-only
Type: brocade-security-type:ipfilter-action-type
Value: activate = This activates the policy. **clone** = This clones the policy.

Optional: Yes

clone-destination-policy-name

Description: The destination IP filter policy name for the clone operation. You can set this parameter with a PATCH request; however, a GET request does not fetch this parameter.
Flag: write-only
Type: brocade-security-type:ipfilter-name-type
Value:
Value: 1 to 20 alphanumeric and underscore characters. The policy name must be unique. The IP filter policy names 'default_ipv4' and 'default_ipv6' are reserved for the default IP filter policies.
Optional: Yes

ipfilter-rule

Description: The IP filter rule configurations.
Flag: read-write
Key: policy-name
This list has the following leafs:

policy-name

Description: The IP filter policy name.
Flag: read-write
Type: brocade-security-type:ipfilter-name-type
Value: 1 to 20 alphanumeric and underscore characters. The policy name must be unique. Note that policy names are case-insensitive and are always stored as lower case. The IP filter policy names 'default_ipv4' and 'default_ipv6' are reserved for the default IP filter policies.
Optional: No

index

Description: The position of the IP filter entry in the IP filter table.
Flag: read-write
Type: uint16
Value: 1 to the current maximum rule number plus one.
Optional: No

source-ip

Description: The source IP address. Note that the source IP option is optional for FORWARD traffic.
Flag: read-write
Type: brocade-security-type:host-type
Value: A valid IP address.
For an IPv4 filter policy, the source address must be a 32-bit IPv4 address in dot decimal notation. The group prefix must be a CIDR block prefix representation. For example, 208.130.32.0/24 represents a 24-bit IPv4 prefix starting from the most significant bit. The special prefix 0.0.0.0/0 matches any IPv4 address. In addition, the keyword **any** is supported to represent any IPv4 address.
For an IPv6 filter policy, the source address must be a 128-bit IPv6 address, in a format acceptable in RFC 3513. The group prefix must be a CIDR block prefix representation. For example, 12AB:0:0:CD30::/64 represents a 64-bit IPv6 prefix starting from the most significant bit. In addition, the keyword **any** is supported to represent any IPv6 address.
Optional: Yes

destination-start-port

Description: The destination port number or the starting port number for a port number range.
Flag: read-write
Type: inet:port-number
Value: 1 to 65535.

Optional: No

destination-end-port

Description: The ending destination port number of the port number range. If no destination end port is specified during IP filter rule creation, the destination end port value is equal to destination start port.
Flag: read-write
Type: inet:port-number
Value: 1 to 65535.
Optional: Yes

protocol

Description: The protocol type (tcp or udp).
Flag: read-write
Type: brocade-security-type:ipfilter-protocol-type
Value: A valid protocol type. Supported types are **tcp** or **udp**. The default is **udp**.
Optional: Yes

permission

Description: The permit or deny action associated with this rule. You can permit or deny (block) ports from 1 through 65535.
Flag: read-write
Type: brocade-security-type:ipfilter-permission-type
Value: permit = Applies the permit action. **deny** = Applies the deny action.
Optional: No

traffic-type

Description: The type of traffic allowed for the specified IP address.
Flag: read-write
Type: brocade-security-type:ipfilter-traffic-type
Value: input = Manages traffic on IP management interfaces. Input traffic is the default. **forward** = Manages bidirectional traffic between the external Ethernet interface (eth0/bond0) and the inband management interface (inbd+).
Optional: Yes

destination-ip

Description: The destination IP address. Note that the destination IP address is optional for INPUT traffic.
Flag: read-write
Type: brocade-security-type:host-type
Value: A valid IP address.
For an IPv4 filter policy, the source address has to be a 32-bit IPv4 address in dot decimal notation. The group prefix has to be a CIDR block prefix representation. For example, 208.130.32.0/24 represents a 24-bit IPv4 prefix starting from the most significant bit. The special prefix 0.0.0.0/0 matches any IPv4 address. In addition, the keyword **any** is supported to represent any IPv4 address.
For an IPv6 filter policy, the source address has to be a 128-bit IPv6 address, in a format acceptable in RFC 3513. The group prefix has to be a CIDR block prefix representation. For example, 12AB:0:0:CD30::/64 represents a 64-bit IPv6 prefix starting from the most significant bit. In addition, the keyword **any** is supported to represent any IPv6 address.
Optional: Yes

user-specific-password-cfg

Description: The password policy configuration for a specific user.
Flag: read-write
Key: user-name

This list has the following leafs

user-name

Description: The user name for a specific user account.
Flag: read-write
Type: brocade-security-type:user-config-user-name-type
Value: 1 to 32 printable ASCII characters.
Optional: Yes

minimum-password-age

Description: The minimum number of days that must elapse before a password can be changed. When the maximum-password-age 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. If a minimum password age value is not configured, a GET request does not fetch this parameter.
Flag: read-write
Type: int32
Value: 0 to 999 days. The default is **0**.
Optional: Yes

maximum-password-age

Description: The maximum number of days that can elapse before a password can be changed. This is the password expiration period. Setting this parameter to 0 disables password expiration. If a maximum password age value is not configured, a GET request does not fetch this parameter.
Flag: read-write
Type: int32
Value: 0 to 999 days. The default is **0**.
Optional: Yes

warn-on-expire

Description: The number of days prior to password expiration that a warning about password expiration displays. If a warn on expire value is not configured, a GET request does not fetch this parameter.
Flag: read-write
Type: int32
Value: 0 to 999 days. The default is **0**.
Optional: Yes

enforce-expire

Description: When activated (set to true), the password for the specified user expires when the expiration date is reached. The user is prompted to change the password on the next successful login. You can set this parameter with a PATCH request; however, a GET request does not fetch this parameter.
Flag: read-write
Type: boolean
Value: true = Password expiration is enforced. **false** = Password expiration is not enforced.
Optional: Yes

hash-type

Description: The hash type (such as md5, sha256, or sha512).
Flag: read-write
Type: brocade-security-type:hash-algorithm-type
Config: false
Value: md5 = The md5 hash type. **sha256** = The sha256 hash type. **sha512** = The sha512 hash type.
Optional: Yes

password-cfg

Description: The generic password policy configuration for all user accounts.
Flag: read-write
The container has the following leafs.

minimum-length

Description: The minimum length of the password. The minimum length of the password can be set from 8 to 40 characters.
Flag: read-write
Type: int32
Value: 8 to 40.
Optional: Yes

character-set

Description: The minimum length on the character set (upper and lowercase letters and special characters) to use in the password. This value must be less than or equal to the minimum-length value.
Flag: read-write
Type: int32
Value: 0 to 40. The default is **0**.
Optional: Yes

user-name-allowed

Description: 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 the username is 'testuser', the password 'testuser123' and 'resutset567' are not allowed.
Flag: read-write
Type: boolean
Value: true = The validation check is activated. **false** = The validation check is deactivated. The default is **true**.
Optional: Yes

minimum-lower-case-character

Description: 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 parameter.
Flag: read-write
Type: int32
Value: 0 to 40. The default is **0**.
Optional: Yes

minimum-upper-case-character

Description: 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 parameter.
Flag: read-write
Type: int32
Value: 0 to 40. The default is **0**.
Optional: Yes

minimum-numeric-character

Description: The minimum number of numeric digits that must appear in the password. The maximum value must be less than or equal to the minimum-length value.
Flag: read-write
Type: int32
Value: 0 to 40. The default is **0**.
Optional: Yes

minimum-special-character

Description: The minimum number of punctuation characters that must appear in the password. All displayable, non-alphanumeric punctuation characters, except the colon (:) are allowed. The maximum value must be less than or equal to the minimum-length value.
Flag: read-write
Type: int32
Value: 0 to 40. The default is **0**.
Optional: Yes

past-password-history

Description: The number of past password values that are disallowed when setting a new password.
Flag: read-write
Type: int32
Value: 0 to 24. The default is **1**.
Optional: Yes

minimum-password-age

Description: The minimum number of days that must elapse before a password can be changed. When the maximum-password-age parameter 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.
Flag: read-write
Type: int32
Value: 0 to 999 days. The default is **0**.
Optional: Yes

maximum-password-age

Description: The maximum number of days that must elapse before a password can be changed, and is also known as the password expiration period. Setting this parameter to zero disables password expiration.
Flag: read-write
Type: int32
Value: 0 to 999 days. The default is **0**.
Optional: Yes

warn-on-expire

Description: The number of days prior to password expiration that a warning about password expiration displays.
Flag: read-write
Type: int32
Value: 0 to 999 days. The default is **0**.
Optional: Yes

lock-out-threshold

Description: 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 zero disables the lockout mechanism.
Flag: read-write
Type: int32
Value: 0 to 999 days. The default is **0**.
Optional: Yes

lock-out-duration

Description: The time, in minutes, after which a previously locked account automatically unlocks. Setting this to zero disables lockout duration, and requires an administrative action to unlock the account.

Flag: read-write
Type: int32
Value: 0 to 99999 minutes. The default is **30**.
Optional: Yes

admin-lock-out-enabled
Description: Whether the lockout policy is enabled for the admin account.
Flag: read-write
Type: boolean
Value: true = Admin lockout is enabled for the admin account. **false** = Admin lockout is not enabled for the admin account.
Optional: Yes

repeat-character-limit
Description: 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.
Flag: read-write
Type: int32
Value: 0 to 40. The default is **1**.
Optional: Yes

sequence-character-limit
Description: 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 of 'passABCword' is disallowed because it contains the sequence 'ABC'.
Flag: read-write
Type: int32
Value: 0 to 40. The default is **1**.
Optional: Yes

password-config-changed
Description: The account password policy status.
Flag: read-write
Type: boolean
Value: true = The account password policy has been changed from the default configuration.
false = The account password policy is the default configuration. The default is **false**
Optional: Yes

reverse-user-name-allowed
Description: The validation check to determine if the password is an exact reverse string of the username.
Flag: read-write
Type: boolean
Value: true = The validation check is activated. **false** = The validation check is deactivated.
Optional: Yes

hash-type
Description: The hash type (md5, sha258, or sha512). This parameter is available only when the password action parameter is configured to hash config (password-action=hash-config).
Flag: read-write
Type: brocade-security-type:hash-algorithm-type

Value: md5 = The md5 hash type. **sha256** = The sha256 hash type. **sha512** = The sha512 hash type.
Optional: Yes

manual-hash-enabled
Description: A password change due to a change in hash type can be manual or enforced at login. If it is configured to be manual, you must change the password now. This parameter is available only when the hash type parameter is configured.
Flag: read-write
Type: boolean
Value: true = The password change due to a change in hash type is manual. **false** = The password change due to a change in hash type is enforced at login.
Optional: Yes

enforce-expire
Description: The expiration of the password for all users except the root user. If activated, the user is prompted for a password change on the next successful login. You can set this parameter with a PATCH request; however, a GET request does not fetch this parameter.
Flag: write-only
Type: boolean
Value: true = Password expiration is enforced. **false** = Password expiration is not enforced.
Optional: Yes

minimum-difference
Description: The number of character differences expected between the old and new password.
Flag: read-write
Type: uint16
Value: 0 to 40.
Optional: Yes

password-action
Description: The configuration or deletion of password policies. You can set this parameter with a PATCH request; however, a GET request does not fetch this parameter.
Flag: write-only
Type: brocade-security-type:password-cfg-operation-type
Value: hash-config = Sets the hash type. Valid values are md5, sha256, or sha512. **default** = Resets all password policies to their default value (sha512). **delete-all** = Removes the password expiration policies of all users.
Optional: Yes

user-config
Description: The user accounts on a switch.
In a Virtual Fabric-enabled environment, you can configure the account's username, its role, and the logical fabrics that the account may access.
An account can have different roles for different logical fabrics. An account can access multiple logical fabrics, but only one logical fabric at a time.
In a logical fabric environment, you can additionally define access to chassis-level commands. An account can have one role in the logical fabric, and another role regarding chassis commands.
Flag: read-write
Key: *name*
This list contains the following leafs:

name
Description: The login name of the account to be created or modified. The name must be unique and must begin with an alphabetic character. User names are case-sensitive and can contain up to 32 alphanumeric characters, including periods (.) and underscores (_). **NOTE:**

Flag: read-write
Type: brocade-security-type:user-config-user-name-type
Value: The login name.
Optional: Yes

password

Description: The password for the account. You can set this parameter with a POST or PATCH request; however, a GET request does not fetch this parameter. **NOTE:** When you assign passwords using FOS REST API, Unicode characters are counted as 2 characters and must be taken into consideration for password length.
Flag: write-only
Type: brocade-security-type:user-password-type
Value: The 8 to 40 character account password. Note that the password must be encoded with the Base64 encoding scheme.
Optional: Yes

role

Description: The account's role for all Logical Fabrics provided with the Logical Fabric list. You can assign any role (user-defined or default) to the account. When you create a user account in an Logical Fabric-enabled environment, you can specify only one role for the user. This role is associated with each of the Logical Fabric IDs.
Flag: read-write
Type: brocade-security-type:user-config-role-type
Value: 4 to 16 alphanumeric characters that specify the account's role.
Optional: Yes

account-description

Description: A printable ASCII string that specifies the description for the new account. To include spaces, place the description in double quotation marks. Note that colons are not permitted.
Flag: read-write
Type: string
Value: 1 to 32 printable ASCII characters.
Optional: Yes

account-enabled

Description: Whether the account is enabled or disabled. Note that once an account is disabled, the sessions associated with the account are terminated.
Flag: read-write
Type: boolean
Value: true = The account is enabled. **false** = The account is disabled.
Optional: Yes

password-change-enforced

Description: Whether an expired password must be changed the first time the user logs into a new or modified account.
Flag: read-write
Type: boolean
Value: true = The expired password must be changed the first time the user logs into a new or modified account. **false** = The expired password does not need to be changed the first time the user logs into a new or modified account.
Optional: Yes

account-locked

Description: Whether the user account is locked after several attempts to login with an invalid password.

Flag: read-write
Type: boolean
Value: true = The user account is locked after several attempts to login with an invalid password.
false = The user account is not locked after several attempts to login with an invalid password.
Optional: Yes

access-start-time
Description: The time, in 24 hour format, during which users can access the switch through Telnet, SSH, console, or Web.
Flag: read-write
Type: string
Value: The time, in 24 hour format, during which users can access the switch through Telnet, SSH, console, or Web.
Optional: Yes

access-end-time
Description: The time, in 24 hour format, during which users can no longer access the switch through Telnet, SSH, console, or Web. This parameter is available only when the access start time parameter (access-start-time) is configured.
Flag: read-write
Type: string
Value: The time, in 24 hour format, during which users can no longer access the switch through Telnet, SSH, console, or Web.
Optional: Yes

virtual-fabric-role-id-list
Description: A list of Virtual Fabrics roles and IDs to be added to the user account.
Flag: read-write
This container contains the following leaf:

role-id
Description: The Virtual Fabrics to be added to the user account.
Flag: read-write
Type: brocade-security-type:virtual-fabric-role-id-type
Value: A valid Virtual Fabric role ID.
Optional: Yes

auth-spec
Description: Replaces the configuration with the specified AAA service.
Flag: read-write
This container contains the following leafs:

authentication-mode
Description: The authentication mode for RADIUS, TACACS+, and LDAP.
Flag: read-write
Type: brocade-security-type:aaa-authspec-type
Value: One of the following supported authentication modes:
local (default) = 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.
Optional: Yes

activate-no-log-out
Description: Whether a change in the authentication mode results in existing sessions being logged out automatically. This parameter is available only when the authentication mode parameter (authentication-mode) is configured. You can set this parameter with a PATCH request; however, a GET request does not fetch this parameter.
Flag: write-only
Type: boolean
Value: true = There is no effect on existing sessions regardless of the chosen authentication mode. **false**(default) = Terminates all existing sessions when the new authentication mode is one of the following: RADIUS only, LDAP only, TACACS+ only, or Local.
Optional: Yes

primary-auth-log-messages
Description: Whether log messages for authentication failure display. This parameter is available only when the authentication mode parameter (authentication-mode) is configured.
Flag: read-write
Type: boolean
Value: true = Log messages display for authentication failure. **false** = Log messages do not display for authentication failure. The default is **true**.
Optional: Yes

radius-server
Description: The RADIUS server configuration.
Flag: read-write
Key: server
This container has the following leafs:

server
Description: 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, where you specify one by server name and the other by IP address.
Flag: read-write
Type: inet:host
Value: A valid IP address or server name.
Optional: Yes

port
Description: The RADIUS server port number.

Flag: read-write
Type: inet:port-number
Value: A valid port number. The default RADIUS server port number is **1812**.
Optional: Yes

secret

Description: The common secret between the switch and the RADIUS server. You can set this parameter with a POST or PATCH request; however, a GET request does not fetch this parameter.
Flag:write-only
Type: string
Value: 8 to 40 alphanumeric characters. The default secret is sharedsecret. The secret must be encoded with the Base64 encoding scheme.
Optional: Yes

timeout

Description: The response timeout for the RADIUS server.
Flag: read-write
Type: brocade-security-type:aaa-timeout-type
Value: 1 to 30 seconds. The default response timeout is 3 seconds.
Optional: Yes

authentication

Description: The remote authentication protocol for the RADIUS server.
Flag: read-write
Type: union {brocade-security-type:aaa-protocols-type; string})
Value: Valid protocols include **pap**, **chap** (default), and **peap-mschapv2**.
Optional: Yes

encryption-type

Description: The encryption algorithm type for the RADIUS servers sharedsecret value.
Flag: read-write
Type: brocade-security-type:aaa-encryption-algorithm-type
Value: Valid algorithm types include **none** and **aes256**. The default is **none**.
Optional: Yes

position

Description: The position to which you want to move the RADIUS server. For example, if you change the server position to 2 (where the old position was 5), the server is moved to position 2 and any existing server at position 2 is moved to position 3.
Flag: read-write
Type: uint16
Value: 1 to 5.
Optional: Yes

tacacs-server

Description: The TACACS+ server configuration.
Flag: read-write
Key: server
This container has the following leafs:

server

Description: The IP address or server name in dot-decimal notation (including IPv6 hexadecimal quadruplets 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, where you specify one by server name and the other by IP address.
Flag: read-write
Type: inet:host
Value: A valid IP address or server name.
Optional: Yes

port
Description: The TACACS+ server port number.
Flag: read-write
Type: inet:port-number
Value: A valid port number. The default TACACS+ server port number is 49.
Optional: Yes

secret
Description: The common secret between the switch and the TACACS+ server. You can set this parameter with a POST or PATCH request; however, a GET request does not fetch this parameter.
Flag:write-only
Type: string
Value: 8 to 40 alphanumeric characters. The default secret is sharedsecret. The secret must be encoded with the Base64 encoding scheme.
Optional: Yes

timeout
Description: The response timeout for the TACACS+ server.
Flag: read-write
Type: brocade-security-type:aaa-timeout-type
Value: 1 to 30 seconds. The default response timeout is 3 seconds.
Optional: Yes

authentication
Description: The remote authentication protocol for the TACACS+ server.
Flag: read-write
Type: union (brocade-security-type:aaa-protocols-type and string)
Value: Valid protocols include **pap** and **chap** (default).
Optional: Yes

encryption-type
Description: The encryption algorithm type for the TACACS+ servers sharedsecret value.
Flag: read-write
Type: brocade-security-type:aaa-encryption-algorithm-type
Value: Valid algorithm types include **none** and **aes256**. The default is **none**.
Optional: Yes

position
Description: The position to which you want to move the TACACS+ server. For example, if you set the position to 2 for a server (where the old position was 5), the server is moved to position 2 and any existing server at position 2 is moved to position 3.
Flag: read-write
Type: uint16
Value: 1 to 5.
Optional: Yes

ldap-server
Description: The LDAP server configuration.
Flag: read-write

Key: server
This container has the following leafs:

server
Description: The IP address or server name in dot-decimal notation (including IPv6 hexadecimal quadruplets 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, where you specify one by server name and the other by IP address.
Flag: read-write
Type: inet:host
Value: A valid IP address or server name.
Optional: Yes

port
Description: The LDAP server port number.
Flag: read-write
Type: inet:port-number
Value: A valid port number. The default LDAP server port number is 389.
Optional: Yes

domain
Description: The name of the active directory domain. For example, brocade.com
Flag: read-write
Type: inet:domain-name
Value: The fully qualified domain name or IP address. The default domain is **local**.
Optional: Yes

timeout
Description: The response timeout for the LDAP server.
Flag: read-write
Type: brocade-security-type:aaa-timeout-type
Value: 1 to 30 seconds. The default response timeout is 3 seconds.
Optional: Yes

tls-mode
Description: The Transport Layer Security (TLS) mode for the LDAP server.
Flag: read-write
Type: brocade-security-type:ldap-tls-mode-type
Value: starttls = Enter to use StartTLS. ldaps = Enter to use LDAPS.
Optional: Yes

position
Description: The position to which you want to move the LDAP server. For example, if you set the position to 2 for a server (where the old position was 5), the server is moved to position 2 and any existing server at position 2 is moved to position 3.
Flag: read-write
Type: uint16
Value: 1 to 5.
Optional: Yes

ldap-role-map
Description: A list of the LDAP AD server role to default switch role mappings.
Flag: read-write
Key: ldap-role
This list contains the following leafs:

ldap-role

Description: The LDAP role to be mapped to a default switch role. The role must be a valid AD server role.
Flag: read-write
Type: string
Value: A valid AD server role.
Optional: Yes

switch-role

Description: The default switch role to be mapped to an LDAP role.
For GET requests, this parameter also contains the virtual fabric role ID. The format is either a string containing the switch user type or the switch user type with the virtual fabric role ID separated by a semicolon (;). For example a string can be **user** (switch user type) or **user=1-10;admin=11-128** (switch user type with the virtual fabric role ID separated by a semicolon)
Flag: read-write
Type: string
Value: A valid default switch role.
Optional: Yes

home-virtual-fabric

Description: The account's home Virtual Fabric. If the ldap-role-map parameter value is not configured, a GET request fetches a 0 value.
Flag: read-write
Type: home-virtual-fabric-type
Value: A valid Virtual Fabric.
Optional: Yes

chassis-access-role

Description: The account's access permissions regarding chassis-level commands.
Flag: read-write
Type: user-config-role-type
Value: A valid role for the account.
Optional: Yes

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.
Flag: read-write
This container has the following leafs:

template-name

Description: The template name.
Flag: read-write
Type: brocade-security-type:sec-crypto-cfg-template-name-type
Value: The template name.
Optional: Yes

action

Description: The operation you want to perform.
Flag: read-write
Type: brocade-security-type:seccrypto-cfg-actions-type
Value: apply = Activates a default or user-defined template file. **verify** = Verifies the running configuration against a required configuration specified in the template file. **import** = Imports

a template file from a specified external host. **export** = Exports a template file to the specified external host.
Note that the template name should not be same as any of the default template names when importing or exporting the file. The default templates are as follows: default_generic, default_strong, default_fips, and default_cc.
Optional: Yes

file-transfer-protocol-type
Description: The file transfer protocol (SCP, SFTP, or FTP).
Flag: read-write
Type: brocade-security-type:sec-crypto-cfg-file-transfer-protocol-type
Value: One of the following valid file transfer protocols: **scp**, **sftp**, or **ftp**.
Optional: Yes

sec-crypto-cfg-template
Description: A list of available templates.
Flag: read-only
Key: name
Config: false
This list has the following leafs:

name
Description: The template name.
Flag: read-only
Type: brocade-security-type:sec-crypto-cfg-template-name-type
Value: The template name.
Optional: Yes

template
Description: The template content.
Flag: read-only
Type: string
Value: The template content.
Optional: Yes

sec-crypto-cfg
Description: The active sec-crypto configurations.
Flag: read-only
Config: false
This container has the following leafs:

ssh-cipher
Description: The active SSH cipher configuration.
Flag: read-only
Type: brocade-security-type:default-string-type
Value: The active SSH cipher configuration.
Optional: Yes

ssh-kex
Description: The active SSH kex configuration.
Flag: read-only
Type: brocade-security-type:default-string-type
Value: The active SSH kex configuration.
Optional: Yes

ssh-mac
Description: The active SSH mac configuration.

Flag: read-only
Type: brocade-security-type:default-string-type
Value: The active SSH mac configuration.
Optional: Yes

https-cipher
Description: The active HTTPS cipher configuration.
Flag: read-only
Type: brocade-security-type:default-string-type
Value: The active HTTPS cipher configuration.
Optional: Yes

radius-cipher
Description: The active RADIUS cipher configuration.
Flag: read-only
Type: brocade-security-type:default-string-type
Value: The active RADIUS cipher configuration.
Optional: Yes

ldap-cipher
Description: The active LDAP cipher configuration.
Flag: read-only
Type: brocade-security-type:default-string-type
Value: The active LDAP cipher configuration.
Optional: Yes

syslog-cipher
Description: The active syslog cipher configuration.
Flag: read-only
Type: brocade-security-type:default-string-type
Value: The active syslog cipher configuration.
Optional: Yes

https-tls-protocol
Description: The active HTTPS TLS protocol.
Flag: read-only
Type: brocade-security-type:tls-protocol-type
Value: The active HTTPS TLS protocol.
Optional: Yes

radius-tls-protocol
Description: The active RADIUS TLS protocol.
Flag: read-only
Type: brocade-security-type:tls-protocol-type
Value: The active RADIUS TLS protocol.
Optional: Yes

ldap-tls-protocol
Description: The active LDAP TLS protocol.
Flag: read-only
Type: brocade-security-type:tls-protocol-type
Value: The active LDAP TLS protocol.
Optional: Yes

syslog-tls-protocol
Description: The active Syslog TLS protocol.
Flag: read-only

Type: brocade-security-type:tls-protocol-type
Value: The active Syslog TLS protocol.
Optional: Yes

x509v3-validation-mode

Description: 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

• 'Certificate Sign' is not mandatory for 'KeyUsage' in a CA certificate.
• OCSP validation for identity certificates is not performed.
• 'BasicConstraints' field is not mandatory in a CA certificate.
• 'CA:True' value is not mandatory in a CA certificate.
• 'Digital Signature' is not mandatory for 'KeyUsage' in a identity certificate.
• Audit logs for establishing and terminating TLS session are not printed.
• SSH sessions with a switch, the 1GB data transfer limit is not enforced for rekeying.
• For RADIUS, LDAP, or Syslog secure authentication using a certificate, a hostname mismatch error is ignored.
• For RADIUS, LDAP, or Syslog secure authentication using a certificate, hostname validation is not performed.
• 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

• 'Certificate Sign' is mandatory for 'KeyUsage' in a CA certificate.
• OCSP validation for identity certificates is performed.
• 'BasicConstraints' field is mandatory in a CA certificate.
• 'CA:True' value is mandatory in a CA certificate.
• 'Digital Signature' is mandatory for 'KeyUsage' in a identity certificate.
• 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.
• In SSH sessions with a switch, the 1GB data transfer limit is enforced for rekeying.
• 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).
• 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.

Flag: read-only
Type: brocade-security-type:validation-mode-type
Value: The X509v3 certificate validation mode type: **Basic** or **Strict**.
Optional: Yes

sshutil

Description: The allowed user configuration. This container is used to configure or display the allowed user and to configure or display the SSH rekey interval.
Flag: read-write
This container has the following leafs:

allow-user-name

Description: The user name. User names are case-sensitive and can contain up to 32 alphanumeric characters, including periods (.) and underscores (_).
Flag: read-write
Type: brocade-security-type:user-config-user-name-type
Value: A valid user name.

Optional: Yes

rekey-interval

Description: The rekey duration in seconds.
Flag: read-write
Type: uint16
Value: **900** to **3600** seconds. **0** to define no rekey interval.
Optional: Yes

sshutil-key

Description: A list of host, public, and private keys. This container is used to generate or delete host keys and public/private keypairs for outgoing SSH connections. This container is also used to display the switch host keys.
Flag: read-write
Keys: algorithm-type; key-type
This list contains the following leafs:

algorithm-type

Description: The algorithm type (rsa, dsa, or ecdsa).
Flag: read-write
Type: brocade-security-type:sshutil-algorithm-type
Value: The algorithm type: **rsa**, **dsa**, or **ecdsa**.
Optional: Yes

key-type

Description: The SSH utility key type (public-private-key or host-key).
Flag: read-write
Type: brocade-security-type:sshutil-key-type
Value: The SSH utility key type: **public-private-key** or **host-key**.
Optional: Yes

passphrase

Description: The password for generating a key. A strong passphrase is 10 to 30 characters long, fairly complex and difficult to guess and contains a mix of upper and lowercase letters, numbers, and nonalphanumeric characters. This parameter is available only when the key type parameter is set to public-private-key (key-type=public-private-key).
Flag: write-only
Type: string
Value: 0 or 5 to 21242 alphanumeric characters. The passphrase must be encoded with the Base64 encoding scheme.
Optional: Yes

size

Description: The size of the Host key.
Flag: read-only
Type: uint16
Value: 0 to 2048.
Config: false
Optional: Yes

fingerprint

Description: The host key fingerprint installed in the switch.
Flag: read-only
Type: string
Value: 1 to 1024 alphanumeric characters.
Config: false
Optional: Yes

sshutil-public-key

Description: A list of public keys. You can delete the public keys for a specified user or all users.
Flag: read-write
Key: user-name
This list contains the following leafs:

user-name

Description: The user name. User names are case-sensitive and can contain up to 32 alphanumeric characters, including periods (.) and underscores (_).
Flag: read-write
Type: brocade-security-type:user-config-user-name-type
Value: A user name. If you specify \"all\" as the user name, the public keys for all user accounts are deleted.
Optional: Yes

public-key

Description: The public key installed in the switch.
Flag: read-only
Type: string
Value: The public key installed in the switch.
Config: false
Optional: Yes

sshutil-public-key-action

Description: The public key import and export container.
Flag: read-write
This container has the follwoing leafs:

action

Description: The import or export action to be performed on the public key. Imported public keys are used for passwordless incoming SSH connections. Exported public keys are used for passwordless outgoing SSH connections.
Flag: read-write
Type: brocade-security-type:sshutil-operation-type
Value: import = Imports a public key from a remote host to the local switch for a specified user.
export = Exports a public key from a local switch to a remote host for a specified user.
Optional: Yes

algorithm-type

Description: The algorithm type (rsa, dsa, or ecdsa). The algorithm type is only available during export (action=export).
Flag: read-write
Type: brocade-security-type:sshutil-algorithm-type
Value: The algorithm type: **rsa**, **dsa**, or **ecdsa**.
Optional: Yes

user-name

Description: The user name. The user name is only available during import (action=import).
Flag: read-write
Type: brocade-security-type:user-config-user-name-type
Value: A user name.
Optional: Yes

remote-user-name

Description: The user name for the host.
Flag: read-write
Type: string

Value: The user name for the host .
Optional: Yes

remote-host-ip
Description: The remote host IP address.
Flag: read-write
Type: string
Value: The remote host IP address.
Optional: Yes

remote-user-password
Description: The password for the remote user.
Flag: read-write
Type: string
Value: The password for the remote user. The password must be encoded with Base64 encoding scheme.
Optional: Yes

remote-directory
Description: The remote directory fully qualified path name.
Flag: read-write
Type: string
Value: The remote directory fully qualified path name.
Optional: Yes

public-key-name
Description: The name of the file in which the public key is stored on the remote host. This is a user-generated file name that must have a .pub extension. The public key name is only available during import (action=import).
Flag: read-write
Type: string
Value: The public key file name.
Optional: Yes

password
Description: The user account password container.
Flag: read-write
This container has the following leafs:

user-name
Description: 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.
Flag: read-write
Type: brocade-security-type:user-config-user-name-type
Value: A user name.
Optional: No

old-password
Description: The old password for the user.
Flag: read-write
Type: brocade-security-type:user-password-type
Value: The old password for the user. The password must be encoded with Base64 encoding scheme.
Optional: No

new-password

Description: The new password for the user. The password must be encoded with the Base64 encoding scheme. Note that new password is mandatory parameter for changing the password.
NOTE: When you assign passwords using FOS REST API, Unicode characters are counted as 2 characters and must be taken into consideration for password length.
Flag: read-write
Type: brocade-security-type:user-password-type
Value: 1 to 40 characters encoded with the Base64 encoding scheme.
Optional: Yes

security-certificate-generate

Description: Generates a self-signed web certificate for HTTPS support. Creates Certificate Signing Requests (CSR) for HTTPS, FCAP, Commoncert, LDAP, Radius and Syslog. **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.
Flag: read-write
This container has the following leafs:

certificate-entity

Description: The certificate entity.
Flag: read-write
Type: brocade-security-type:gen-certificate-entity-type
Value: csr = The certificate-signing request entity. **ca-client** = The certificate authority (CA) client entity. **ca-server** = The certificate authority (CA) server entity. **switch** = The switch certificate entity.
Optional: No

certificate-type

Description: The certificate type.
Flag: read-write
Type: brocade-security-type:certificate-application-type
Value: commoncert = The common certificate. **https** = The certificate for secure HTTP. **radius** = The certificate for Remote Authentication Dial-In User Service. **ldap** = The certificate for Lightweight Directory Access Protocol. **syslog** = The certificate for syslog. **fcap** = The certificate for Fibre Channel Authentication Protocol.
Optional: Yes

algorithm-type

Description: The algorithm type (rsa, dsa, or ecdsa).
Flag: read-write
Type: brocade-security-type:sshutil-algorithm-type
Value: The algorithm type: **rsa**, **dsa**, or **ecdsa**. The default is **rsa**.
Optional: Yes

key-size

Description: The size of the key in bytes. The greater the value, the more secure is the connection; however, performance degrades with size. Specifying larger key sizes (4096+) may require more switch processing time than some third-party REST applications have configured for request timeout, resulting in a 500 Internal Server Error result. Also, generating a larger key size may trigger CPU usage alerts in MAPS.
Flag: read-write
Type: brocade-security-type:keysize-type
Value: 1024, **2048**, **4096**, or **8192** bytes. The default is **2048** bytes.
Optional: Yes

hash-type

Description: The hash type.
Flag: read-write
Type: brocade-security-type:seccertmgmt-hash-type
Value: sha1 = This Secure Hash Algorithm (SHA) generates a 160-bit (20-byte) hash value. **sha256** = This SHA generates a unique, fixed size 256-bit (32-byte) hash. **sha512** = This SHA uses 64 byte words. The default is **sha256**.
Optional: Yes

years

Description: The number of years for which the certificate is valid.
Flag: read-write
Type: uint16
Value: 1 to 50 years. The default is **5** years.
Optional: Yes

country-name

Description: The country name. This parameter is available only when the certificate entity is configured as CSR (certificate-entity=CSR). Also note that if this parameter is present in a self-signed certificate, it is ignored.
Flag: read-write
Type: string
Value: The country name.
Optional: Yes

state-name

Description: The state name. This parameter is available only when the certificate entity is configured as CSR (certificate-entity=CSR). Also note that if this parameter is present in a self-signed certificate, it is ignored.
Flag: read-write
Type: string
Value: The state name.
Optional: Yes

locality-name

Description: The locality name. This parameter is available only when the certificate entity is configured as CSR (certificate-entity=CSR). Also note that if this parameter is present in a self-signed certificate, it is ignored.
Flag: read-write
Type: uint16
Value: The .
Optional: Yes

organization-name

Description: The organization name. This parameter is available only when the certificate entity is configured as CSR (certificate-entity=CSR). Also note that if this parameter is present in a self-signed certificate, it is ignored.
Flag: read-write
Type: string
Value: The organization name.
Optional: Yes

unit-name

Description: The unit name. This parameter is available only when the certificate entity is configured as CSR (certificate-entity=CSR). Also note that if this parameter is present in a self-signed certificate, it is ignored.

Flag: read-write
Type: string
Value: The unit name.
Optional: Yes

domain-name
Description: The fully qualified domain name or IP address. This parameter is available only when the certificate entity is configured as CSR (certificate-entity=CSR). Also note that if this parameter is present in a self-signed certificate, it is ignored.
Flag: read-write
Type: string
Value: The fully qualified domain name or IP address.
Optional: Yes

security-certificate-action
Description: The import or export action of third-party certificates on a switch, including Public Key Infrastructure (PKI) based certificates, LDAP certificates, RADIUS certificates, and syslog CA certificates.
Flag: read-write
This container has the following leafs:

protocol
Description: The protocol type for import or export.
Flag: read-write
Type: brocade-security-type:seccertmgmt-protocol-type
Value: scp = The secure copy protocol. **ftp** = The file transfer protocol.
Optional: Yes

certificate-entity
Description: The certificate entity.
Flag: read-write
Type: brocade-security-type:gen-certificate-entity-type
Value: csr = The certificate-signing request entity. **ca-client** = The Certification Authority (CA) client entity. **ca-server** = The CA server entity. **switch** = The switch certificate entity.
Optional: Yes

certificate-type
Description: The certificate type.
Flag: read-write
Type: brocade-security-type:certificate-application-type
Value: commoncert = The common certificate. **https** = The certificate for secure HTTP. **radius** = The certificate for Remote Authentication Dial-In User Service. **ldap** = The certificate for Lightweight Directory Access Protocol. **syslog** = The certificate for syslog. **fcap** = The certificate for FCAP. **all** = All certificates.
Optional: Yes

certificate-name
Description: The certificate name.
Flag: read-write
Type: string
Value: 5 to 128 alphanumeric characters plus hyphens, periods, and underscore characters.
Optional: Yes

operation
Description: The security certificate operations (such as import or export).
Flag: read-write
Type: brocade-security-type:seccertmgmt-operation-type

Value: 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 CA that issues the certificate.
Optional: Yes

security-certificate
Description: A list of security certificates on a switch, including PKI-based certificates, LDAP certificates, RADIUS certificates, syslog CA certificates, and FCAP certificates.
Flag: read-only
Key: certificate-entity; certificate-type
This list has the following leafs:

certificate-entity
Description: The certificate entity.
Flag: read-only
Type: brocade-security-type:gen-certificate-entity-type
Value: csr = The certificate-signing request entity. **ca-client** = The Certification Authority (CA) client entity. **ca-server** = The CA server entity. **switch** = The switch certificate entity.
Optional: Yes

certificate-type
Description: The certificate type.
Flag: read-only
Type: brocade-security-type:certificate-application-type
Value: commoncert = The common certificate. **https** = The certificate for secure HTTP. **radius** = The certificate for Remote Authentication Dial-In User Service. **ldap** = The certificate for Lightweight Directory Access Protocol. **syslog** = The certificate for syslog. **fcap** = The certificate for FCAP. **all** = All certificates.
Optional: Yes

certificate
Description: The CSR, CA-client, CA-server, or switch certificates installed on the switch.
Flag: read-only
Type: string
Value: The CSR, CA-client, CA-server, or switch certificates installed on the switch.
Optional: Yes

certificate-hexdump
Description: The CSR, CA-client, CA-server, or switch certificates, in raw hex format, installed on the switch.
Flag: read-only
Type: string
Value: The CSR, CA-client, CA-server, or switch certificates, in raw hex format, installed on the switch.
Optional: Yes

Supported Methods

Only the OPTIONS, HEAD, GET, PATCH, DELETE, and POST operations are supported in this module.

History

Release version	History
Fabric OS 8.2.1	This API call was introduced.
Fabric OS 8.2.3a	This API called was modified to add the tls-mode parameter.

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 IP filter policy configuration by specifying the IP filter policy in the request (for example, `GET <base_URI>/running/brocade-security/ipfilter-policy/fwd_ipv4`).

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.

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 IP filter policy configuration by specifying the IP filter policy in the request (for example, GET *<base_URI>*/running/brocade-security/ipfilter-rule/fwd_ipv4).

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>
        <traffic-type>INPUT</traffic-type>
        <destination-ip>any</destination-ip>
    </ipfilter-rule>
    <ipfilter-rule>
        <policy-name>default_ipv4</policy-name>
        <index>2</index>
        <source-ip>any</source-ip>
        <destination-start-port>23</destination-start-port>
        <destination-end-port>23</destination-end-port>
        <protocol>tcp</protocol>
        <permission>permit</permission>
        <traffic-type>INPUT</traffic-type>
        <destination-ip>any</destination-ip>
    </ipfilter-rule>
.
.
.
    <ipfilter-rule>
        <policy-name>default_ipv6</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>
        <traffic-type>INPUT</traffic-type>
        <destination-ip>any</destination-ip>
    </ipfilter-rule>
    <ipfilter-rule>
        <policy-name>default_ipv6</policy-name>
        <index>2</index>
        <source-ip>any</source-ip>
        <destination-start-port>23</destination-start-port>
        <destination-end-port>23</destination-end-port>
        <protocol>tcp</protocol>
        <permission>permit</permission>
```

```
            <traffic-type>INPUT</traffic-type>
            <destination-ip>any</destination-ip>
        </ipfilter-rule>
    .
    .
    .
        <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>
            <traffic-type>FORWARD</traffic-type>
            <destination-ip>192.26.1.3</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>
            <traffic-type>FORWARD</traffic-type>
            <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>
    <traffic-type>FORWARD</traffic-type>
    <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>Test1</user-name>
            <hash-type>SHA512</hash-type>
        </user-specific-password-cfg>
        <user-specific-password-cfg>
            <user-name>Test123456789012345678901 2</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>
        <home-virtual-fabric>128</home-virtual-fabric>
        <virtual-fabric-role-id-list>
            <role-id>root=1-128</role-id>
        </virtual-fabric-role-id-list>
        <chassis-access-role>root</chassis-access-role>
        <access-start-time/>
        <access-end-time/>
    </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>
        <home-virtual-fabric>128</home-virtual-fabric>
        <virtual-fabric-role-id-list>
            <role-id>admin=1-128</role-id>
        </virtual-fabric-role-id-list>
        <chassis-access-role>admin</chassis-access-role>
        <access-start-time/>
        <access-end-time/>
    </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>
        <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/>
        <access-start-time/>
        <access-end-time/>
    </user-config>
    <user-config>
        <name>testuser</name>
        <role>user</role>
        <account-description/>
        <account-enabled>true</account-enabled>
        <password-change-enforced>false</password-change-enforced>
        <account-locked>false</account-locked>
        <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/>
        <access-start-time/>
        <access-end-time/>
    </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>
    <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>
    <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/>
    <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>
   <position>3</position>
</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>10.10.10.10</server>
        <port>389</port>
        <domain>local</domain>
        <timeout>3</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://10.10.10.11/rest/running/brocade-security/ldap-server
```

Request Body

```
<ldap-server>
    <server>10.10.10.11</server>
    <port>389</port>
    <domain>local</domain>
    <timeout>3</timeout>
    <tls-mode>starttls</tls-mode>
    <position>1</position>
</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 Default 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</switch-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_name*

URI Request

```
DELETE https://10.10.10.10/rest/running/brocade-security/ldap-role-map/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_fips</name>
        <template>[Ver]  0.1
[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
[FIPS]
SelfTests:yes
BootProm:yes
Zeroize:yes
Inside:yes
[X509v3]
Validation:Basic
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_cc</name>
        <template>[Ver]  0.1
[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
[X509v3]
Validation:Strict
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_strong</name>
        <template>[Ver]  0.1
[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-sha2-256,hmac-sha2-512
[AAA]
RAD_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!SSLv3
LDAP_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM:!SSLv3
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:!SSLv3
Protocol:TLSv1.2
[X509v3]
Validation:Basic
</template>
    </sec-crypto-cfg-template>
    <sec-crypto-cfg-template>
        <name>default_generic</name>
        <template>[Ver]  0.1
[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-md5,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:Any
LDAP_Protocol:Any
[LOG]
Syslog_Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
Syslog_Protocol:Any
[HTTPS]
Ciphers:!ECDH:!DH:HIGH:-MD5:!CAMELLIA:!SRP:!PSK:!AESGCM
Protocol:Any
[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-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>
        <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 Configuration

The following example uses the GET request to display the allowed user 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 Authentication

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>
        <passphrase/>
        <size>1024</size>
```

```
            <fingerprint>03:78:fc:7a:99:f2:05:b0:99:cb:b5:cf:30:8d:1c:1b</fingerprint>
        </sshutil-key>
        <sshutil-key>
            <algorithm-type>dsa</algorithm-type>
            <key-type>host-key</key-type>
            <passphrase/>
            <size>1024</size>
            <fingerprint>8c:8a:bc:00:e3:82:de:09:ea:56:fc:d6:92:c3:49:31</fingerprint>
        </sshutil-key>
        <sshutil-key>
            <algorithm-type>ecdsa</algorithm-type>
            <key-type>host-key</key-type>
            <passphrase/>
            <size>256</size>
            <fingerprint>be:62:31:01:52:3c:b7:4e:38:d2:53:e4:1f:98:7a:79</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>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 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
- 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
- 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>
    <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.

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>
```

brocade-snmp

This module provides a detailed view of the SNMP configurations that are 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.

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 SNMP Module 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:def-control
       |  +--rw security-get-level?       snmp-types:security-level
       |  +--rw security-set-level?       snmp-types:security-level
       |  +--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
       |  +--ro community-group?          snmp-types:group-name
       +--rw v1-trap* [index]
       |  +--rw index                     uint16
       |  +--rw host?                     inet:host
       |  +--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
       |  +--ro user-group?               snmp-types:group-name
       |  +--rw authentication-protocol?  snmp-types:authentication-protocol
       |  +--rw privacy-protocol?         snmp-types:privacy-proto
       |  +--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?                     inet:host
```

```
|  +--rw trap-severity-level?       snmp-types:severity-level
|  +--rw port-number?               inet:port-number
|  +--rw informs-enabled?           boolean
+--rw access-control* [index]
   +--rw index                      uint16
   +--rw host?                      inet:ip-address
   +--rw access-level?              snmp-types:access-permission
```

URI Format

The URI format for this module takes the following form:

- *<base_URI>*/running/brocade-snmp/system/ followed by the leafs as listed in the module tree.
- *<base_URI>*/running/brocade-snmp/mib-capability/ followed by the leafs as listed in the module tree.
- *<base_URI>*/running/brocade-snmp/trap-capability/ followed by the leafs as listed in the module tree.
- *<base_URI>*/running/brocade-snmp/v1-account/ followed by the leafs as listed in the module tree.
- *<base_URI>*/running/brocade-snmp/v1-trap/ followed by the leafs as listed in the module tree.
- *<base_URI>*/running/brocade-snmp/v3-account/ followed by the leafs as listed in the module tree.
- *<base_URI>*/running/brocade-snmp/v3-trap/ followed by the leafs as listed in the module tree.
- *<base_URI>*/running/brocade-snmp/access-control/ followed by the leafs as listed in the module tree.

Parameters

brocade-snmp

Description: The container for SNMP query and trap notification configuration.
Flag: read-write
This container has the following leafs:

system

Description: The system-wide SNMP configurations.
Flag: read-write
Key: *<key>*
This container has the following leafs:

description

Description: A printable ASCII string that provides a description of the entity.
Flag: read-write
Type: string
Value: 4 to 255 printable ASCII characters that provide a description of the entity. This parameter should include the full name and version identification of the system's hardware type, software operating system, and networking software. Refer to RFC 1213.
Optional: Yes

location

Description: The physical location of this node.
Flag: read-write
Type: string
Value: The physical location of this node. Refer to RFC 1213.
Optional: Yes

contact

Description: The contact person for this managed node, together with how to contact this person.
Flag: read-write
Type: string
Value: The contact person for this managed node, together with how to contact this person. Refer to RFC 1213.
Optional: Yes

informs-enabled

Description: Indicates whether informs is enabled or disabled in the switch.
Flag: read-write
Type: boolean
Value: true = Informs is enabled in the switch. **false** = Informs is disabled in the switch.
Optional: Yes

encryption-enabled

Description: Indicates whether password encryption is enabled or disabled. If password encryption is enabled, the authentication and privacy passwords are stored as encrypted. If password encryption is disabled, authentication and privacy passwords are stored as plain text.
Flag: read-write
Type: boolean
Value: true = Password encryption is enabled in the switch. **false** = Password encryption is disabled in the switch.
Optional: Yes

audit-interval

Description: The SNMP audit interval in minutes.
Flag: read-write
Type: uint16
Value: 1 to 1440.
Optional: Yes

default-config

Description: List of SNMP configurations.
Flag: read-write
This container has the following leaf:

default-control

Description: The SNMP configuration (such as snmpv1, snmpv3, access control, system group, mib capability, and audit interval).
Flag: read-write
Type: snmp-types:def-control
Value: The SNMP configuration (such as snmpv1, snmpv3, access control, system group, mib capability, and audit interval).
Optional: Yes

security-get-level

Description: The SNMP security access level for the GET operation.
Flag: read-write
Type: snmp-types:security-level
Value: 0 = No Security. **1** = Authentication only. **2** = Authentication and privacy. **3** = No access.
Optional: Yes

security-set-level

Description: The SNMP security access level for the SET operation.
Flag: read-write

Type: snmp-types:security-level
Value: 0 = No Security. **1** = Authentication only. **2** = Authentication and privacy. **3** = No access.
Optional: Yes

snmpv1-enabled
Description: Indicates whether SNMPv1 is enabled or disabled in the switch.
Flag: read-write
Type: boolean
Value: true = SNMPv1 is enabled in the switch. **false** = SNMPv1 is disabled in the switch.
Optional: Yes

mib-capability
Description: A list of MIBs.
Key: mib-name
Flag: read-write
This list has the following leafs:

mib-name
Description: The MIB name.
Flag: read-write
Type: snmp-types:mibs-name
Value: The MIB name.
Optional: Yes

is-mib-enabled-state
Description: Indicates whether the MIB is enabled or disabled.
Flag: read-write
Type: boolean
Value: true = The MIB is enabled. **false** = The MIB is disabled.
Optional: Yes

trap-capability
Description: A list of traps.
Key: trap-name
Flag: read-write
This container has the following leafs:

trap-name
Description: The trap name.
Flag: read-write
Type: snmp-types:traps-name
Value: The trap name.
Optional: Yes

is-trap-enabled-state
Description: Indicates whether the trap is enabled or disabled.
Flag: read-write
Type: boolean
Value: true = The trap is enabled. **false** = The trap is disabled.
Optional: Yes

severity
Description: The trap recipient severity level. This parameter is available only when the trap name is swEventTrap (trap-name=swEventTrap). 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 and connUnitEventTrap), are sent to configured trap recipients.

Flag: read-write
Type: snmp-types:severity-level
Value: The trap recipient severity level (none, critical, error, warning, informational, and debug).
Optional: Yes

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.
Key: index
Flag: read-write
This list has the following leafs:

index

Description: The label for this object.
Flag: read-write
Type: uint16
Value: 1 thorugh 6.
Optional: Yes

community-name

Description: The community string.
Flag: read-write
Type: string
Value: The community string.
Optional: Yes

community-group

Description: Indicates whether the SNMPv1 community belongs to a read-only or a read-write group.
Flag: read-only
Type: snmp-types:group-name
Config: false
Value: ro = The group is read-only. **rw** = The group is read-write.
Optional: Yes

v1-trap

Description: The SNMPv1 trap notification. Refer to RFC 3413.
Key: index
Flag: read-write
This list has the following leafs:

index

Description: The label for this object.
Flag: read-write
Type: uint16
Value: 1 thorugh 6.
Optional: Yes

host

Description: The IP address of the trap recipient system.
Flag: read-write
Type: inet:host
Value: A valid IP address.
Optional: Yes

trap-severity-level

Description: 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 and connUnitEventTrap) are sent to configured trap recipients.
Flag: read-write
Type: snmp-types:severity-level
Value: The trap recipient severity level (none, critical, error, warning, informational, and debug).
Optional: Yes

port-number

Description: The UDP port where SNMP traps are received.
Flag: read-write
Type: inet:port-number
Value: A valid UDP port where SNMP traps are received.
Optional: Yes

v3-account

Description: The SNMPv3 user account. This parameter is used to access system via SNMPv3 in a secure 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.
Key: index
Flag: read-write
This list has the following leafs:

index

Description: The label for this object.
Flag: read-only
Type: uint16
Value: 1 thorugh 6.
Optional: Yes

user-name

Description: The name of the user that connects to the agent.
Flag: read-write
Type: string
Value: 2 through 32 character name of the user that connects to the agent.
Optional: Yes

user-group

Description: Indicates whether the SNMPv3 user belongs to a read-only or a read-write group.
Flag: read-only
Type: snmp-types:group-name
Config: false
Value: **ro** = The group is read-only. **rw** = The group is read-write.
Optional: Yes

authentication-protocol

Description: The authorization protocol (MD5 or SHA) for the SNMPv3 user. Refer to RFC 3414.
Flag: read-write
Type: snmp-types:authentication-protocol-type
Value: MD5 or SHA.
Optional: Yes

privacy-protocol

Description: The privacy protocol (DES or AES-128) for the SNMPv3 user. Refer to RFC 3414.
Flag: read-write
Type: snmp-types:privacy-protocol-type
Value: DES or AES-128
Optional: Yes

authentication-password

Description: The authentication password for the SNMPv3 user to access the system resources. The password should be base64 encoded. Refer to RFC 3414.
Flag: read-write
Type: string
Value: 8 to 32 characters. The password must be encoded with the Base64 encoding scheme.
Optional: Yes

privacy-password

Description: The privacy password for the SNMPv3 user to access the system resources. The password should be base64 encoded. Refer to RFC 3414.
Flag: read-write
Type: string
Value: 8 to 32 characters. The password must be encoded with the Base64 encoding scheme.
Optional: Yes

manager-engine-id

Description: The user-defined engine ID for the SNMP manager which is used to receive the SNMPv3 informs notifications. This parameter is only applicable when the informs is enabled (informs-enabled).
Flag: read-write
Type: string
Value: 5 to 32 character user-defined engine ID for the SNMP manager.
Optional: Yes

v3-trap

Description: The SNMPv3 trap notification. Refer to RFC 3413.
Key: trap-index
Flag: read-write
This container has the following leafs:

trap-index

Description: The label for this object.
Flag: read-write
Type: uint16
Value: 1 thorugh 6.
Optional: Yes

usm-index

Description: The index of the user account associated with the trap index.
Flag: read-write
Type: uint16
Value: 1 thorugh 6.
Optional: Yes

host

Description: The IP address of trap recipient system.

Flag: read-write
Type: inet:host
Value: A valid IP address.
Optional: Yes

trap-severity-level

Description: 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 and connUnitEventTrap) are sent to configured trap recipients.
Flag: read-write
Type: snmp-types:severity-level
Value: The trap recipient severity level (none, critical, error, warning, informational, and debug).
Optional: Yes

port-number

Description: The UDP port where SNMP traps are received.
Flag: read-write
Type: inet:port-number
Value: A valid UDP port where SNMP traps are received.
Optional: Yes

informs-enabled

Description: Indicates whether the informs is enabled or disabled.
Flag: read-write
Type: uint8
Value:
Value: true = Informs is enabled. **false** = Informs is disabled.
Optional: Yes

access-control

Description: The SNMP access control list to restrict SNMP GET, SET, and trap operations to the hosts under an host subnet area.
Key: index
Flag: read-write
This container has the following leafs:

index

Description: The label for this object.
Flag: read-write
Type: uint16
Value: 0 thorugh 5.
Optional: Yes

host

Description: The subnet area of the access host. The IP address, for which SNMP operations works only for the hosts configured in the ACL list. The IP address supports both IPv4 and IPv6 addresses.
Flag: read-write
Type: inet:ip-address
Value: A valid IP address.
Optional: Yes

access-level

Description: The access level of the SNMP access control entry. The access level can be either read-only or read-write.

Flag: read-write
Type: snmp-types:access-permission
Value: ro = The access level is read-only. **rw** = The access level is read-write.
Optional: Yes

Supported Methods

Only the OPTIONS, GET, PATCH, and HEAD, operations are supported in this module.

Examples

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.

```
<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>
        <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>2</security-get-level>
    <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>swIPv6ChangeTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swPmgrEventTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swFabricReconfigTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swFabricSegmentTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swExtTrap</trap-name>
        <is-trap-enabled-state>false</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swStateChangeTrap</trap-name>
        <is-trap-enabled-state>false</is-trap-enabled-state>
```

```
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swPortMoveTrap</trap-name>
        <is-trap-enabled-state>false</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swBrcdGenericTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swDeviceStatusTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>swZoneConfigChangeTrap</trap-name>
        <is-trap-enabled-state>false</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>connUnitStatusChange</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>connUnitEventTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>connUnitPortStatusChange</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>linkRNIDDeviceRegistration</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>linkRNIDDeviceDeRegistration</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>linkLIRRListenerAdded</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
```

```
        <trap-name>linkLIRRListenerRemoved</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>linkRLIRFailureIncident</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>fruStatusChanged</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>cpStatusChanged</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>fruHistoryTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>linkDown</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>linkUp</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>mapsTrapAM</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>mapsQuietTimeExpirationTrap</trap-name>
        <is-trap-enabled-state>true</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>t11ZsRequestRejectNotify</trap-name>
        <is-trap-enabled-state>false</is-trap-enabled-state>
        <severity>none</severity>
    </trap-capability>
    <trap-capability>
        <trap-name>t11ZsMergeFailureNotify</trap-name>
        <is-trap-enabled-state>false</is-trap-enabled-state>
        <severity>none</severity>
```

```
    </trap-capability>
    <trap-capability>
        <trap-name>t11ZsMergeSuccessNotify</trap-name>
        <is-trap-enabled-state>false</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>rw</community-group>
        <community-name>Secret C0de</community-name>
    </v1-account>
    <v1-account>
        <index>2</index>
        <community-group>rw</community-group>
        <community-name>OrigEquipMfr</community-name>
    </v1-account>
    <v1-account>
        <index>3</index>
        <community-group>rw</community-group>
        <community-name>private</community-name>
    </v1-account>
    <v1-account>
        <index>4</index>
        <community-group>ro</community-group>
        <community-name>public</community-name>
    </v1-account>
    <v1-account>
        <index>5</index>
        <community-group>ro</community-group>
        <community-name>common</community-name>
    </v1-account>
    <v1-account>
        <index>6</index>
        <community-group>ro</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>
```

Configure 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>0.0.0.0</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>rw</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>rw</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>3</index>
        <user-name>snmpadmin3</user-name>
        <user-group>rw</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>4</index>
        <user-name>snmpuser1</user-name>
        <user-group>ro</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>5</index>
        <user-name>snmpuser2</user-name>
        <user-group>ro</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>ro</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 '10:10:10:00:00:00:00:00:00'.

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</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>10.38.38.165</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>10.38.38.90</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>6</usm-index>
   <host>10.38.38.90</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.

History

Release version	History
Fabric OS 8.2.1b	This API call was introduced.

brocade-time

This module provides enables you to view and configure the time zone and clock server.

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-types: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-types:ts-ntp-type
     |  +--ro active-server?        brocade-time-types:ts-ntp-type
     |  +--rw ts-auth-spec?         brocade-time-types: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-types:ts-key-type
        +--rw key?                  string
```

URI Format

The URI format for this module takes one of the following forms:

• *<base_URI>*`/running/brocade-time/time-zone` followed by the leafs as listed in the module tree to view or configure the time zone parameters.

• *<base_URI>*`/running/brocade-time/clock-server` followed by the leafs as listed in the module tree to view or configure the clock server parameters.

• *<base_URI>*`/running/brocade-time/ntp-clock-server` followed by the leafs as listed in the module tree to view or configure the Network Time Protocol (NTP) clock server.

• *<base_URI>*`/running/brocade-time/ntp-clock-server-key` followed by the leafs as listed in the module tree to view or configure the NTP clock server symmetric keys.

Supported Methods

Only the OPTIONS, HEAD, GET, and PATCH operations are supported in this module.

History

Release Version	History
Fabric OS 8.2.1	This API call was introduced.
Fabric OS 9.1.0	This API call was modified to add the ts-auth-spec and ts-legacy-mode leafs and the ntp-clock-server and ntp-clock-server-key lists. This API call was modified to edit the name and active-server leafs and the clock-server container.

Examples

This section provides examples for the brocade-time module.

brocade-zone

This module is used for managing Fibre Channel zoning.

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
       |  +--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
       |  |  +--rw zone-type?                zone-type-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
          +--rw cfg-name?                   zoning-name-type
          +--rw checksum?                   string
          +--rw cfg-action?                 uint8
          +--rw default-zone-access?        uint8
          +--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 enabled-zone* [zone-name]
             +--ro zone-name                zoning-name-type
             +--ro zone-type?               zone-type-type
             +--ro member-entry
                +--ro entry-name*           union
                +--ro principal-entry-name* union
```

URI Format

The URI format for this module takes the following form:

• *<base_URI>*`/running/brocade-zone/defined-configuration` followed by the leafs as listed in the module tree.

• *<base_URI>*`/running/brocade-zone/effective-configuration` followed by the leafs as listed in the module tree.

URI Parameter Definitions

- *cfg-name*—The zone configuration name
- *alias-name*—The zone alias name
- *zone-name*—The zone name

NOTE
If all the switches in the fabric and both the CPs in a chassis system are running Fabric OS 8.1.0 or later, all of these names can start with a number or a letter, and may contain a hyphen (-), underscore (_), dollar sign ($), or a caret (^), except as the first character. If you merge a domain running Fabric OS 8.0.1 or lower with a domain having zone object names (configuration, alias, or zone name) starting with numbers, or containing "$", "-", or "^", the fabric will segment.

NOTE
In certain circumstances, zone, zone configuration, and zone alias names that use the dollar sign character ($) may be problematic in a REST request. The solution is to replace the dollar sign with its encoded version (%24).

Parameters

NOTE
The top-level container name changed from "zoning" to "brocade-zone". The previous top-level container name "zoning" is still supported in this release.

brocade-zone

Description: This is the primary container for this module.
Flag: read-write
This container has the following containers:

defined-configuration

Description: The defined zoning configuration.
NOTE
If you perform a GET request on the defined-configuration container and there is an open zone transaction whose user is logged out of the REST session, the open zone transaction is aborted.
Flag: read-write
Key: *cfg-name*
This container has the following leafs:

cfg

Description: The name of the interface configuration. There may only be one.
Flag: read-write
Type: string
Values: A list of configuration names.
Optional: No
This list has the following leafs:

cfg-name

Description: The name of the zone configuration.
Flag: read-write
Type: zoning-name-type
Values: 1 to 64 alphanumeric characters, as well as the $, ^, - (hyphen), and _ (underscore) characters. This is case-sensitive, and must not contain special (high-ASCII) characters. See notes about names above.
Optional: No

member-zone

Description: List of configuration member zones.
Flag: read-write

Optional: No
This list has the following leafs:

zone-name
Description: The zone name.
Flag: read-write
Type: zoning-name-type
Values: 1 to 64 alphanumeric characters, as well as the $, ^, - (hyphen), and _ (underscore) characters. This is case-sensitive, and must not contain special (high-ASCII) characters. See notes about names above.
Optional: No

zone
Description: The list of zones. The members can only be identified as a WWN, domain,index, or zone alias.
Flag: read-write
Key: *zone-name*
This list has the following leafs:

zone-name
Description: The zone name.
Flag: read-write
Type: zoning-name-type
Values: 1 to 64 alphanumeric characters, as well as the $, ^, - (hyphen), and _ (underscore) characters. This is case-sensitive, and must not contain special (high-ASCII) characters. See notes about names above.
Optional: No

zone-type
Description: The zone type. Note that target-created peer zone (TDZ) types cannot be created or modified (only deleted). Also note that this parameter must be supplied for any user-created peer zone operation.
Flag: read-write
Type: zone-type-type
Optional: Yes
Values: 0 = default. (The non-peer zone type that is already set on the switch.) 1 = User-created peer zone. 2 = Target-created peer zone. Default: 0
Optional: Yes

member-entry
Description: The zone member.
Flag: read-write
This list has the following leafs:

entry-name
Description: A list of the members in the zone. The members can only be identified as a WWN, domain,index, or zone alias.
Flag: read-write
Type: zone-member-type
Values: A valid zone alias, a WWN, or a port identifier in the format “domain,index”.
Optional: No

principal-entry-name
Description: A list of the principal members in the peer zone. The members can only be identified as a WWN, domain,index, or zone alias.
Flag: read-write

Type: zone-member-type
Values: A valid zone alias, a WWN, or a port identifier in the format “domain,index”.
Optional: This field is optional if the zone-type field is not specified or is set to 0. It is required if the “zone-type” field is set to 1 (user peer) and a write is being performed.

alias

Description: The list of zone aliases.
Flag: read-write
Key: *alias-name*
This list has the following leafs:

alias-name

Description: The zone alias name.
Flag: read-write
Type: zoning-name-type
Values: 1 to 64 alphanumeric characters, as well as the $, ^, - (hyphen), and _ (underscore) characters. This is case-sensitive, and must not contain special (high-ASCII) characters. See notes about names above.
Optional: No

member-entry

Description: The alias member.
Flag: read-write
Optional: No
This list has the following leaf:

alias-entry-name

Description: The zone alias.
Flag: read-write
Type: union { brocade:wwn-type; domain-index-type }
Values: A valid WWN or a port identifier in the format “domain,index”.
Optional: No

effective-configuration

Description: The effective zoning configuration.
NOTE If you perform a GET request on the effective-configuration container and there is an open zone transaction whose user is logged out of the REST session, the open zone transaction is aborted.
Flag: read-write
Optional: No
Key: *cfg-name*
This container has the following leafs:

cfg-name

Description: The name of the effective configuration to be enabled or a configuration that is currently enabled.
This must be the name of a configuration that is already defined.
Flag: read-write
Type: zoning-name-type
Values: 1 to 64 alphanumeric characters, as well as the $, ^, - (hyphen), and _ (underscore) characters. This is case-sensitive, and must not contain special (high-ASCII) characters. See notes about names above.
Optional: Yes

checksum

Description: The checksum of the configuration database.

The checksum is an MD5 (Message-Digest algorithm 5) calculation (128-bit cryptographic hash) used to verify data integrity. This resource is used by the client to verify that 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. It must be written when the changes are saved. The changes will be accepted if the checksum agrees and rejected if it does not.
The checksum field is required for `cfgenable` operations and also for *`cfg-action`* values of 1 or 2 (See *`cfg-action`* entry below.)
Flag: read-write
Type: string
Values: 32 hexadecimal characters
Optional: Yes

`cfg-action`
Description: The action to be done to the pending zoning configuration changes.
Flag: read-write
Type: unit8
Values: 0 = Read is not applicable. 1 = Save the pending changes. (Similar to `cfgSave` command.) 2 = Disable the effective configuration. (Similar to `cfgDisable` command.) 3 = Clear the entire zone database. (Similar to `cfgTransAbort` command.) 4 = Clear any pending changes. (Similar to `cfgClear` or `cfgTransAbort` command.) Default: 0
Optional: Yes

`default-zone-access`
Description: Enable or disable zone access.
The default zoning mode controls device access if zoning is not implemented or if there is no enabled zone configuration.
Flag: read-write
Type: unit8
Values: 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 in the fabric (All Access). Default: 1
Optional: Yes

`db-max`
Description: The maximum size of the zone database in bytes. The maximum size for the zone database is the upper limit for the zone-defined configuration determined by the amount of nonvolatile memory available for storing the defined configuration. The maximum size for the zone database is further reduced due to a message header that is propagated with the zone configuration to all switches in the fabric.

> **NOTE**
> If you perform a GET request on the db-max parameter and there is an open zone transaction whose user is logged out of the REST session, the open zone transaction is aborted.

Flag: read-only
Type: uint32
Config: false
Units: bytes
Values: 0 to the maximum zone database size supported by the platforms in the fabric. Chassis-only fabrics support up to 2 MB (2084640 bytes). If one or more fixed-port switches are in a fabric, the maximum is 1 MB (1042320 bytes).
Optional: Yes

`db-avail`
Description: The size in bytes of free space available in the zone database.

NOTE
If you perform a GET request on the db-avail parameter and there is an open zone transaction whose user is logged out of the REST session, the open zone transaction is aborted.

Flag: read-only
Type: uint32
Config: false
Units: bytes
Values: 0 to the maximum zone database size supported by the platforms in the fabric. Chassis-only fabrics support up to 2 MB (2084640 bytes). If one or more fixed-port switches are in a fabric, the maximum is 1 MB (1042320 bytes). Default: 0
Optional: Yes

db-committed

Description: 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.

NOTE
If you perform a GET request on the db-committed parameter and there is an open zone transaction whose user is logged out of the REST session, the open zone transaction is aborted.

Flag: read-only
Type: unit32
Config: false
Units: bytes
Values: 4 to the maximum zone database size supported by the chassis or fabric. Chassis-only fabrics support up to 2 MB (2084640 bytes). If one or more fixed-port switches are in a fabric, the maximum is 1 MB (1042320 bytes). Default: 4
Optional: Yes

db-transaction

Description: The size in bytes of memory bytes required to commit the current transaction. The database transaction size will not be 0 if the defined configuration is being modified by REST, Telnet, API, or other applications; otherwise it is 0.

NOTE
If you perform a GET request on the db-transaction parameter and there is an open zone transaction whose user is logged out of the REST session, the open zone transaction is aborted.

Flag: read-only
Type: unit32
Config: false
Units: bytes
Values: 0 to the maximum zone database size supported by the chassis or fabric. Chassis-only fabrics support up to 2 MB (2084640 bytes). If one or more fixed-port switches are in a fabric, the maximum is 1 MB (1042320 bytes). Default: 0
Optional: Yes

transaction-token

Description: The token for the current zoning transaction.
Flag: read-only
Type: unit32
Config: false

Values: 0 through 4294967280. If a transaction is open, this will be a 32-bit transaction token. If no transaction is open, this will be 0. Default: 0 (No outstanding zoning transaction.)
Optional: Yes

db-chassis-wide-committed
Description: The size in bytes for the zone database across all the logical switches in the chassis if virtual fabrics are in use. If virtual fabrics are not in use, this is the size of the zone database for the switch. The *db-committed* value is smaller than this resource due to storage overhead.
Flag: read-only
Type: unit32
Config: false
Units: bytes
Values: 0 to the maximum zone database size supported by the chassis or fabric. Chassis-only fabrics support up to 2 MB (2092741 bytes). If one or more fixed-port switches are in a fabric, the maximum is 1 MB (1042320 bytes). Default: 0
Optional: Yes

enabled-zone
Description: The enabled zones.
Flag: read-only
Optional: No
Key: *zone-name*
This list has the following leafs:

zone-name
Description: The name of the zone.
Flag: read-only
Type: zoning-name-type
Values: 1 to 64 alphanumeric characters, as well as the $, ^, - (hyphen), and _ (underscore) characters. This is case-sensitive, and must not contain special (high-ASCII) characters. See notes about names above.
Config: false
Optional: No

zone-type
Description: The zone type.
Flag: read-only
Type: zone-type-type
Values: 0 = default (The zone type that is already set on the switch.) 1 = User peer zone. 2 = Target peer zone. Default: 0
Optional: No. On reads, this will always be populated.

member-entry
Description: The zone members.
Flag: read-only
This list has the following leafs:

entry-name
Description: List of the members in the zone.
Zone aliases are expanded when the configuration is enabled and are not included in the effective configuration.
Flag: read-only
Type: union { zoning-name-type: brocade:wwn-type }
Values: A valid WWN or a port identifier in the format “domain,index”.
Optional: No

principal-entry-name

Description: If the zone is a peer zone, this property contains the list of principals in the zone.
The members can be identified using either a WWN or a domain,index ID. Zone aliases are expanded when the configuration is enabled and consequently are not included in the effective configuration.
Flag: read-only
Type: union { brocade:wwn-type; domain-index-type }
Values: A valid WWN or a port identifier in the format “domain,index”.
Optional: No. This leaf will only be returned if the zone is a peer zone. It will not be returned for other zone types.

Supported Methods

The GET, POST, PATCH, DELETE, OPTIONS, and HEAD operations are supported in this module.

Examples

Retrieving the List of Zone Configurations

This example uses a GET request to retrieve the list of all the 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/*input 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>
      <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/*input zone-name*

URI

```
POST http://10.10.10.10/rest/running/brocade-zone/defined-configuration/zone/zone-name/qos
```

Request Body

```
<zone-type>1</zone-type>
<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

DELETE *<base_URI>*/running/brocade-zone/defined-configuration/cfg/cfg-name/*configuration_name*

URI

```
DELETE 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 “200 OK” status in the header.

History

Release Version	History
Fabric OS 8.2.0	This API call was introduced.
Fabric OS 8.2.0a	This API call was updated to include information about running a GET request on the effective-configuration container for the db-max, db-avail, db-committed, or db-transaction parameters.
Fabric OS 8.2.1b	The top-level container name changed from "zoning" to "brocade-zone". The previous top-level container name "zoning" is still supported in this release.

FOS REST API Modules for Extension Features

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

brocade-extension-ip-route

This module is used to retrieve or configure IP route information on the specified switch.

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               uint16
        +--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
```

URI Format

The URI format for this module takes the following form:

<base_URI>/running/brocade-extension-ip-route/extension-ip-route/ followed by the leafs as listed in the module tree to create, modify, or delete an IP route.

Supported Methods

The GET, POST, PATCH, DELETE, OPTIONS, and HEAD operations are supported in this module.

History

Release Version	History
Fabric OS 8.2.0	This API call was introduced.
Fabric OS 9.1.0	This API was modified to edit the ip-address and status-flags parameters.

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.

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
```

URI Format

The URI format for this module takes the following form:

<base_URI>`/running/brocade-extension-ipsec-policy/extension-ipsec-policy/` followed by the leafs as listed in the module tree to create, modify, or delete an IPsec policy.

Supported Methods

Only the GET, POST, PATCH, DELETE, OPTIONS, and HEAD operations are supported in this module.

History

Release Version	History
Fabric OS 8.2.0	This API call was introduced.

brocade-extension-tunnel

This module is used to retrieve information on or configure Brocade Extension tunnels.

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

```
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
     |  +--rw admin-enabled?                          uint8
     |  +--rw fast-write-enabled?                     uint8
     |  +--rw tape-read?                              uint8
     |  +--rw tape-write?                             uint8
     |  +--ro tunnel-status?                          extension:operational-type
     |  +--ro ipsec-enabled?                          uint8
     |  +--rw ipsec-policy?                           string
     |  +--ro ha-operational-status?                  uint16
     |  +--rw ficon?                                  uint8
     |  +--rw ficon-xrc-acceleration?                 uint8
     |  +--rw ficon-tape-write-acceleration?          uint8
     |  +--rw ficon-tape-read-acceleration?           uint8
     |  +--rw ficon-tin-tir-emulation?                uint8
     |  +--rw ficon-device-acknowledgement-emulation? uint8
     |  +--rw ficon-read-block-id-emulation?          uint8
     |  +--rw ficon-teradata-read-acceleration?       uint8
     |  +--rw ficon-teradata-write-acceleration?      uint8
     |  +--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
     |  +--rw ip-extension?                           uint16
     |  +--rw load-level?                             extension:tunnel-load-level
     |  +--ro active-load-level?                      extension:tunnel-load-level
     |  +--ro peer-load-level?                        extension:tunnel-load-level
     |  +--rw compression-tunnel?                     uint16
     |  +--rw compression-protocol
     |  |  +--rw fc-compression?   string
     |  |  +--rw ip-compression?   string
     |  +--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
|  +--ro flow-status?              uint16
|  +--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
|  +--ro admin-status?                 uint8
|  +--ro path-mtu-discovered?          uint32
|  +--rw minimum-communication-rate?   uint32
|  +--rw maximum-communication-rate?   uint32
|  +--rw keep-alive-timeout?           uint32
|  +--ro vlan-id?                      uint32
|  +--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
|  |  +--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
|  +--ro circuit-status?               extension:operational-type
|  +--rw local-ha-ip-address?          inet:ip-address
|  +--rw remote-ha-ip-address?         inet:ip-address
|  +--rw arl-algorithm-mode?           uint32
```

```
     +--ro extension-circuit-statistics* [name circuit-id] {fibrechannel:fibrechannel_extension_platform}?
     |  +--ro name                  extension:ve-interface-type
     |  +--ro circuit-id            uint16
     |  +--ro flow-status?          uint16
     |  +--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 communication-rate?   uint32
     |  +--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
     +--ro wan-statistics* [ve-port circuit-id connection-id] {fibrechannel:fibrechannel_extension_platform}?
        +--ro ve-port                               extension:ve-interface-type
        +--ro circuit-id                            uint16
        +--ro connection-id                         uint32
        +--ro priority?                             extension:extension-qos-type
        +--ro ha-type?                              extension:tunnel-ha-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 operation-mode?                       extension:tcp-operation-mode
+--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 retransmit-timeout?                   yang:zero-based-counter64
+--ro retransmits?                          yang:zero-based-counter64
+--ro maximum-retransmits?                  yang:zero-based-counter64
+--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-total?         yang:zero-based-counter32
+--ro in-queued-out-of-order-maximum?       yang:zero-based-counter32
```

for data type descriptions.

URI Formats

The URI format for this module takes one of the following forms:

• *<base_URI>*/running/brocade-extension-tunnel/extension-tunnel/ followed by the leafs as listed in the module tree to create, modify, or delete an extension tunnel.

• *<base_URI>*/running/brocade-extension-tunnel/extension-circuit/ followed by the leafs as listed in the module tree to create, modify, or delete a circuit on an extension tunnel.

• *<base_URI>*/running/brocade-extension-tunnel/extension-tunnel-statistics/ followed by the leafs as listed in the module tree to return the statistics for the extension tunnel.

• *<base_URI>*/running/brocade-extension-tunnel/extension-circuit-statistics/ followed by the leafs as listed in the module tree to return the statistics for the extension circuit.

• *<base_URI>*/running/brocade-extension-tunnel/circuit-qos-statistics/ followed by the leafs as listed in the module tree to return the circuit QOS statistics for the extension circuit.

• *<base_URI>*/running/brocade-extension-tunnel/wan-statistics/ followed by the leafs as listed in the module tree to return the WAN statistics for the extension tunnel.

Supported Methods

The following table lists the supported and unsupported methods for this module.

Table 21: Supported and unsupported methods

URI	Supported Methods	Unsupported Methods
<base_URI>/running/brocade-extension-tunnel/extension-tunnel	GET, POST, PATCH, DELETE, OPTIONS, and HEAD	
<base_URI>/running/brocade-extension-tunnel/extension-tunnel-statistics	GET, OPTIONS, and HEAD	POST, PATCH, and DELETE
<base_URI>/running/brocade-extension-tunnel/extension-circuit	GET, POST, PATCH, DELETE, OPTIONS, and HEAD	
<base_URI>/running/brocade-extension-tunnel/extension-circuit-statistics	GET, OPTIONS, and HEAD	POST, PATCH, and DELETE

History

Release Version	History
Fabric OS 8.2.0	This API call was introduced.
Fabric OS 8.2.1	This API call was modified to add the out-packet-total and out-packet-lost leafs to the extension-circuit-statistics list. It was also modified to add Brocade 7810 and/or SX6 blade specific changes to the remote-ha-ip-address, compression-tunnel, compression-protocol, circuit-id, minimum-communication-rate, and maximum-communication-rate parameters.
Fabric OS 9.0.0	This API call was modified to add the local-hcl-in-progress, remote-hcl-in-progress, fc-ha-status, ip-ha-status, and last-error to the extension-tunnel-statistics list. This API call was also modified to add the circuit-qos-statistics and wan-statistics lists
Fabric OS 9.0.1	This API call was modified to edit the distribution-value parameter.
Fabric OS 9.1.0	This API call was modified to edit the user-friendly-name, ipsec-policy, fc-compression, ip-compression, distribution, distribution-value, fc-high-qos, fc-medium-qos, fc-low-qos, fc-high-qos, ip-medium-qos, ip-low-qos, path-mtu-discovered, priority-control, fc-priority-high, fc-priority-medium, fc-priority-low, ip-priority-high, ip-priority-medium, and ip-priority-low parameters. This API call was modified to add the peer-product-name, in-bytes, out-bytes, communication-rate, rtt, and time-generated parameters.

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.

> **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
```

URI Format

The URI for this module takes the following form:

<base_URI>`/running/brocade-interface/extension-ip-interface/` followed by the leafs as listed in the module tree.

Supported Methods

The GET, POST, PATCH, DELETE, OPTIONS, and HEAD operations are supported in this module.

History

Release Version	History
Fabric OS 8.2.0	This API call was introduced.
Fabric OS 9.1.0	This API call was modified to edit the ip-address and status-flags leafs.

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.

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
      |  +--rw enabled-state?                uint8
      |  x--rw speed?                        brocade-interface-types:speed-type
      |  +--rw protocol-speed?               fibrechannel:protocol-speed-type
      |  +--ro mac-address?                  yang:mac-address
      |  +--ro operational-status?           uint16
      |  +--rw persistent-disable?           uint8
      |  +--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
```

```
|  +--rw reset-statistics?     uint8
|  +--ro time-generated?       fibrechannel:time-generated-type
```

URI Format

The URI for this module takes one of the following forms:

• *<base_URI>*/running/brocade-interface/gigabitethernet/ to return a list of all the Gigabit Ethernet ports on the switch.

• *<base_URI>*/running/brocade-interface/gigabitethernet-statistics/ to return the statistics for all the Gigabit Ethernet ports on the switch.

Supported Methods

The GET, PATCH, OPTIONS, and HEAD methods are supported in this module. The DELETE and POST methods are not supported.

History

Release Version	History
Fabric OS 8.2.0	This API call was introduced.
Fabric OS 9.1.0	This API call was modified to add the protocol-speed leaf. This API call was modified to edit the auto-negotiation-enabled, portchannel-name, lldp-profile, and lldp-enabled-state leafs. This API call was modified to deprecate the speed leaf (use the protocol-speed leaf).

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.

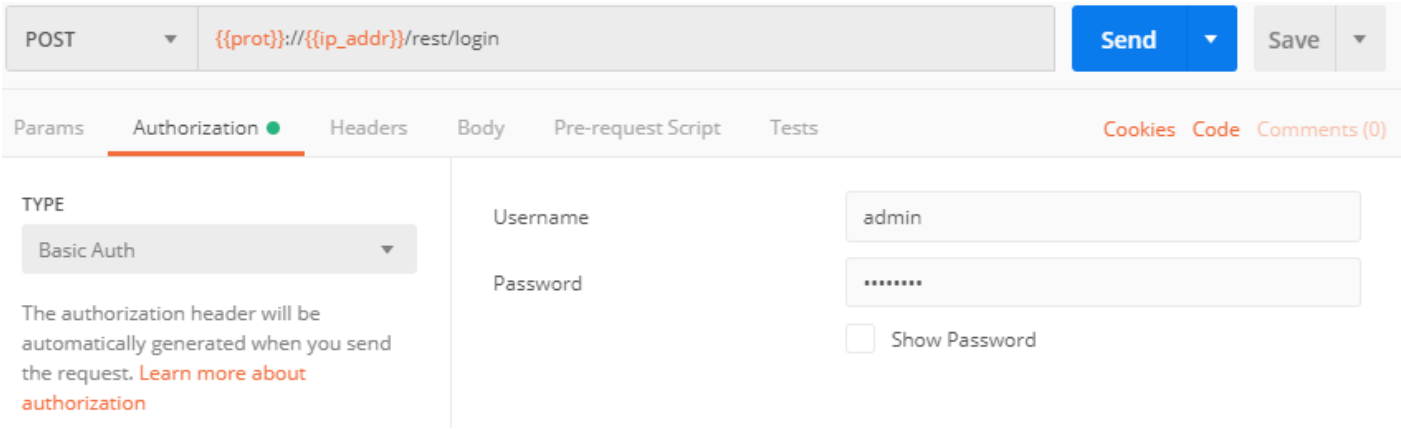
1. Use a POST request to log on to a fabric.

   ```
   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.

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.

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>
           <cfg-action>0</cfg-action>
           <db-max>1045274</db-max>
           <db-avail>1023818</db-avail>
           <db-committed>17186</db-committed>
           <db-transaction>0</db-transaction>
           <db-chassis-wide-committed>18198</db-chassis-wide-committed>
           <transaction-token>0</transaction-token>
           <checksum>fd0d26c940d4f55b467b76552d687e74</checksum>   <== record this value
           <cfg-name>cfg1</cfg-name>
           <enabled-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>
           </enabled-zone>
           <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>
            </member-entry>
        </enabled-zone>
    </effective-configuration>
</Response>
```

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/1
   ```

   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>
        <cfg-action>0</cfg-action>
        <db-max>1045274</db-max>
        <db-avail>1023812</db-avail>
        <db-committed>17192</db-committed>
        <db-transaction>0</db-transaction>
        <db-chassis-wide-committed>18204</db-chassis-wide-committed>
        <transaction-token>0</transaction-token>
        <checksum>d2ba41573853f387d9fd1d76f3b56112</checksum>
        <cfg-name>cfg1</cfg-name>
        <enabled-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>
        </enabled-zone>
        <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>
            </member-entry>
        </enabled-zone>
        <enabled-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>
        </enabled-zone>
    </effective-configuration>
</Response>
```

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/4
   ```

   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/1
   ```

   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 5-minute timeout limit.

Cancellation of Multiswitch Concurrent Transactions

For this example, assume that there are two switches in a fabric (Switch 1 and Switch 2) and the transaction has been pending for more than 5 minutes.

1. A FOS API REST client on Switch 1 (Rest_sw1) starts a zone transaction.
2. While the transaction is in progress, a CLI user on Switch 2 (CLI_sw2) starts a different zone transaction.
3. Subsequently CLI_sw2 commits its zone changes while the Rest_sw1 zone transaction is still open.
   As part of the fabric-wide commit of the CLI_sw2 zone transaction, Rest_sw1's zone transaction will be automatically cancelled.

As Rest_sw1 does not have any knowledge of CLI_sw2's commit operation, if Rest_sw1 attempts to perform another zone edit operation, Rest_sw1 will receive an error similar to the following notifying Rest_sw1 that their previous zone transaction was cancelled:

```
<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>
```

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 5 minutes.
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>
```

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 22: 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

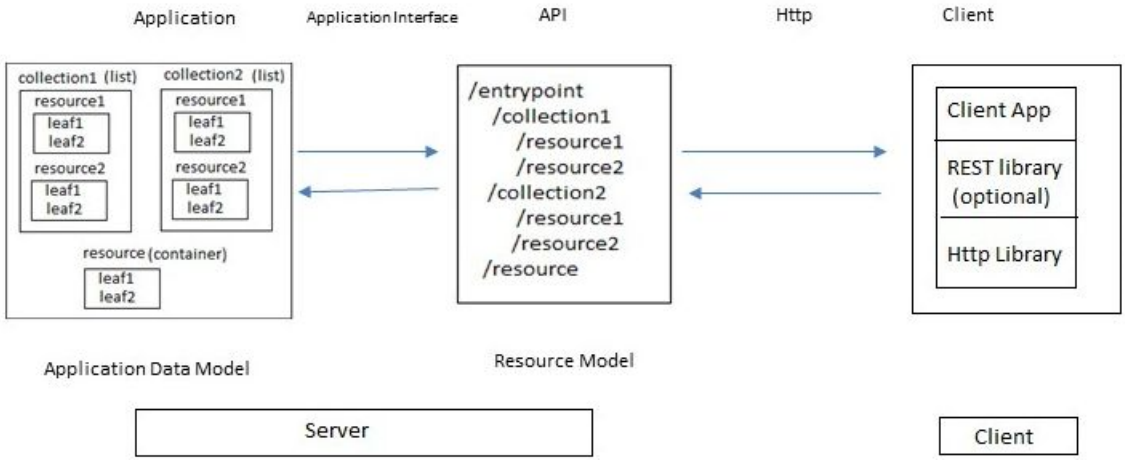
The resource data model is based on the YANG data model. All top-level containment statements such as “List” and “Container” that are present in the YANG data model are the resources, with a few exceptions.

Table 23: 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 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 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 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` will identify 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>
```

Revision History

FOS-82X-REST-API-RM103; 27 December 2021

- In brocade-chassis, added the `entitlement-serial-number` leaf to the `chassis` container.

FOS-82X-REST-API-RM102; 22 June 2021

- In FOS REST API Version History, updated the brocade-security version.
- In brocade-security, added the `tls-mode` parameter.

FOS-82X-REST-API-RM101; 20 December 2019

In PATCH Method, added the following note:

> **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.

FOS-82X-REST-API-RM100; 12 November 2019

- In FOS REST API Described, added support for the FOS session-less REST API.
- In Fabric OS REST API Overview, added Using Brocade FOS REST API Session-Less Operation.
- In Fabric OS REST API Overview, added Scalability Recommendations for FOS REST API Clients.
- In brocade-operation-supportsave, added the `port` parameter to set the SCP or SFTP port number.

FOS-821-REST-API-RM105; 13 February 2019

- FOS REST API Version History is a new section.
- Deprecated and Obsolete Resources is a new section.
- In Additional FOS REST API Data Types, added the SNMP module data types.
- brocade-snmp is a new module.
- brocade-license is a new module.
- The following modules were merged into one module file.

Table 24: Module Name Changes

8.2.1 File Name	8.2.1 Module Name	8.2.1b File Name	8.2.1b Module Name
brocade-fibrechannel.yang	brocade-fibrechannel	brocade-interface.yang	brocade-interface
brocade-extension-ip-interface.yang	brocade-extension-ip-interface		

8.2.1 File Name	8.2.1 Module Name	8.2.1b File Name	8.2.1b Module Name
brocade-gigabitethernet.yang	brocade-gigabitethernet		

- In brocade-interface/fibrechannel, added the `physical-state` and`pod-license-status` parameters. Merged the brocade-fibrechannel, brocade-extension-ip-interface, and brocade-gigabitethernet modules into the brocade-interface module. However, this section covers only brocade-fibrechannel.
- In brocade-interface/extension-ip-interface, merged the brocade-fibrechannel, brocade-extension-ip-interface, and brocade-gigabitethernet modules into the brocade-interface module. However, this section covers only brocade-extension-ip-interface.
- In brocade-interface/gigabitethernet, merged the brocade-fibrechannel, brocade-extension-ip-interface, and brocade-gigabitethernet modules into the brocade-interface module. However, this section covers only brocade-gigabitethernet.
- The following top-level container names have changed.

Table 25: Top-Level Container Name Changes

File Name	Module Name	Existing Top-Level Container	New Top-Level Container
brocade-fabric.yang	brocade-fabric	fabric	brocade-fabric
brocade-fibrechannel-diagnostics.yang	brocade-fibrechannel-diagnostics	diagnostics	brocade-fibrechannel-diagnostics
brocade-fibrechannel-logical-switch.yang	brocade-fibrechannel-logical-switch	logical-switch	brocade-fibrechannel-logical-switch
brocade-fibrechannel-switch.yang	brocade-fibrechannel-switch	switch	brocade-fibrechannel-switch
brocade-zone.yang	brocade-zone	zoning	brocade-zone

- In brocade-fabric, changed the top-level container name from "fabric" to "brocade-fabric". The previous top-level container name "fabric" is still supported in this release. Added the `path-count` parameter to the module.
- In brocade-fibrechannel-diagnostics, changed the top-level container name from "diagnostics" to "brocade-fibrechannel-diagnostics". The previous top-level container name "diagnostics" is still supported in this release.
- In brocade-fibrechannel-logical-switch, changed the top-level container name from "logical-switch" to "brocade-fibrechannel-logical-switch". The previous top-level container name "logical-switch" is still supported in this release.
- In brocade-fibrechannel-switch, changed the top-level container name from "switch" to "brocade-fibrechannel-switch". The previous top-level container name "switch" is still supported in this release.
- In brocade-zone, changed the top-level container name from "zoning" to "brocade-zone". The previous top-level container name "zoning" is still supported in this release.
- In brocade-fru, added the `power-consumption` , `power-usage` , `time-alive` , and `time-awake` parameters.

FOS-821-REST-API-RM104; 27 December 2018

- Changed Brocade Network Advisor to Brocade SANnav Management Portal throughout document.

FOS-821-REST-API-RM103; 4 December 2018

- Updated What Is New in This Document and Revision History.

FOS-821-REST-API-RM102; 17 October 2018

- Updated the copyright statement and Document Feedback.

FOS-821-REST-API-RM101; 28 September 2018

- Updated brocade-security.

FOS-821-REST-API-RM100; 28 August 2018

- Added JSON support throughout the document.
- Updated Fabric OS REST session configuration.
- Updated Additional FOS REST API Data Types.
- Added FOS REST API Modules for Operations.
- Updated brocade-access-gateway.
- Added brocade-chassis.
- Updated brocade-fabric.
- Added brocade-fibrechannel-configuration.
- Added brocade-fibrechannel-trunk.
- Added brocade-fru.
- Updated brocade-fibrechannel.
- Added brocade-logging.
- Added brocade-maps.
- Added brocade-media.
- Added brocade-security.
- Added brocade-time.
- Updated brocade-extension-tunnel.
- Added Creating a Port Trunk Area Using JSON.
- Added Monitoring the Execution of a Rule .
- Added Creating a User-Defined Group and Using the Group in Monitoring a Rule.
- Added Viewing the Switch Status Policy Report.
- Added Generating and Importing a Security Certificate.
- Added Configuring SSH Public Key Authentication on a Switch for Incoming Connections.
- Added Configuring SSH Public Key Authentication on a Switch for Outgoing Connections.
- Added JSON Resource Representation.


BROADCOM®