

RUCKUS FastIron RESTCONF Programmers Guide, 10.0.00

Supporting FastIron Software Release 10.0.00

Copyright, Trademark and Proprietary Rights Information

© 2022 CommScope, Inc. All rights reserved.

No part of this content may be reproduced in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without written permission from CommScope, Inc. and/or its affiliates ("CommScope"). CommScope reserves the right to revise or change this content from time to time without obligation on the part of CommScope to provide notification of such revision or change.

Export Restrictions

These products and associated technical data (in print or electronic form) may be subject to export control laws of the United States of America. It is your responsibility to determine the applicable regulations and to comply with them. The following notice is applicable for all products or technology subject to export control:

These items are controlled by the U.S. Government and authorized for export only to the country of ultimate destination for use by the ultimate consignee or end-user(s) herein identified. They may not be resold, transferred, or otherwise disposed of, to any other country or to any person other than the authorized ultimate consignee or end-user(s), either in their original form or after being incorporated into other items, without first obtaining approval from the U.S. government or as otherwise authorized by U.S. law and regulations.

Disclaimer

THIS CONTENT AND ASSOCIATED PRODUCTS OR SERVICES ("MATERIALS"), ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED. TO THE FULLEST EXTENT PERMISSIBLE PURSUANT TO APPLICABLE LAW, COMMSCOPE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TITLE, NON-INFRINGEMENT, FREEDOM FROM COMPUTER VIRUS, AND WARRANTIES ARISING FROM COURSE OF DEALING OR COURSE OF PERFORMANCE. CommScope does not represent or warrant that the functions described or contained in the Materials will be uninterrupted or error-free, that defects will be corrected, or are free of viruses or other harmful components. CommScope does not make any warranties or representations regarding the use of the Materials in terms of their completeness, correctness, accuracy, adequacy, usefulness, timeliness, reliability or otherwise. As a condition of your use of the Materials, you warrant to CommScope that you will not make use thereof for any purpose that is unlawful or prohibited by their associated terms of use.

Limitation of Liability

IN NO EVENT SHALL COMMSCOPE, COMMSCOPE AFFILIATES, OR THEIR OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, SUPPLIERS, LICENSORS AND THIRD PARTY PARTNERS, BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER, EVEN IF COMMSCOPE HAS BEEN PREVIOUSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, WHETHER IN AN ACTION UNDER CONTRACT, TORT, OR ANY OTHER THEORY ARISING FROM YOUR ACCESS TO, OR USE OF, THE MATERIALS. Because some jurisdictions do not allow limitations on how long an implied warranty lasts, or the exclusion or limitation of liability for consequential or incidental damages, some of the above limitations may not apply to you.

Trademarks

ARRIS, the ARRIS logo, COMMSCOPE, RUCKUS, RUCKUS WIRELESS, the Ruckus logo, the Big Dog design, BEAMFLEX, CHANNELFLY, FASTIRON, ICX, SMARTCELL and UNLEASHED are trademarks of CommScope, Inc. and/or its affiliates. Wi-Fi Alliance, Wi-Fi, the Wi-Fi logo, Wi-Fi Certified, the Wi-Fi CERTIFIED logo, Wi-Fi Protected Access, the Wi-Fi Protected Setup logo, Wi-Fi Protected Setup, Wi-Fi Multimedia and WPA2 and WMM are trademarks or registered trademarks of Wi-Fi Alliance. All other trademarks are the property of their respective owners.

Contents

Preface	5
Contacting RUCKUS Customer Services and Support.....	5
What Support Do I Need?.....	5
Open a Case.....	5
Self-Service Resources.....	6
Document Feedback.....	6
RUCKUS Product Documentation Resources.....	6
Online Training Resources.....	6
Document Conventions.....	7
Notes, Cautions, and Safety Warnings.....	7
Command Syntax Conventions.....	7
About This Document	9
New in This Document	9
Supported Hardware.....	9
RESTCONF for RUCKUS ICX Switches	11
Introduction.....	11
RESTCONF-Supported HTTP Methods.....	13
RESTCONF Supported Modules.....	13
ICX Management.....	14
RESTCONF Resources.....	16
URI.....	16
Base URI.....	16
Data.....	16
YANG Library Version	16
Operation Resource.....	17
CRUD Operations.....	17
XML Representation.....	18
JSON Representation.....	18
Media Types.....	19
Capabilities.....	19
Schema Resources.....	19
YANG Module Retrieval.....	20
Query Parameters.....	20
Transport Protocol Requirements.....	20
Web Requests with Tools (curl).....	21
RESTCONF Configuration.....	21
Configuring ICX Using the RESTCONF Management Interface.....	21
Enabling the RESTCONF Management Interface.....	46
Displaying RESTCONF Configuration and Status Information.....	46
High Availability.....	49
Security Considerations.....	49

Preface

• Contacting RUCKUS Customer Services and Support.....	5
• Document Feedback.....	6
• RUCKUS Product Documentation Resources.....	6
• Online Training Resources.....	6
• Document Conventions.....	7
• Command Syntax Conventions.....	7

Contacting RUCKUS Customer Services and Support

The Customer Services and Support (CSS) organization is available to provide assistance to customers with active warranties on their RUCKUS products, and customers and partners with active support contracts.

For product support information and details on contacting the Support Team, go directly to the RUCKUS Support Portal using <https://support.ruckuswireless.com>, or go to <https://www.commscope.com/ruckus> and select **Support**.

What Support Do I Need?

Technical issues are usually described in terms of priority (or severity). To determine if you need to call and open a case or access the self-service resources, use the following criteria:

- Priority 1 (P1)—Critical. Network or service is down and business is impacted. No known workaround. Go to the **Open a Case** section.
- Priority 2 (P2)—High. Network or service is impacted, but not down. Business impact may be high. Workaround may be available. Go to the **Open a Case** section.
- Priority 3 (P3)—Medium. Network or service is moderately impacted, but most business remains functional. Go to the **Self-Service Resources** section.
- Priority 4 (P4)—Low. Requests for information, product documentation, or product enhancements. Go to the **Self-Service Resources** section.

Open a Case

When your entire network is down (P1), or severely impacted (P2), call the appropriate telephone number listed below to get help:

- Continental United States: 1-855-782-5871
- Canada: 1-855-782-5871
- Europe, Middle East, Africa, Central and South America, and Asia Pacific, toll-free numbers are available at <https://support.ruckuswireless.com/contact-us> and Live Chat is also available.
- Worldwide toll number for our support organization. Phone charges will apply: +1-650-265-0903

We suggest that you keep a physical note of the appropriate support number in case you have an entire network outage.

Self-Service Resources

The RUCKUS Support Portal at <https://support.ruckuswireless.com> offers a number of tools to help you to research and resolve problems with your RUCKUS products, including:

- Technical Documentation—<https://support.ruckuswireless.com/documents>
- Community Forums—<https://community.ruckuswireless.com>
- Knowledge Base Articles—<https://support.ruckuswireless.com/answers>
- Software Downloads and Release Notes—https://support.ruckuswireless.com/#products_grid
- Security Bulletins—<https://support.ruckuswireless.com/security>

Using these resources will help you to resolve some issues, and will provide TAC with additional data from your troubleshooting analysis if you still require assistance through a support case or RMA. If you still require help, open and manage your case at https://support.ruckuswireless.com/case_management.

Document Feedback

RUCKUS is interested in improving its documentation and welcomes your comments and suggestions.

You can email your comments to RUCKUS at #Ruckus-Docs@commscope.com.

When contacting us, include the following information:

- Document title and release number
- Document part number (on the cover page)
- Page number (if appropriate)

For example:

- RUCKUS SmartZone Upgrade Guide, Release 5.0
- Part number: 800-71850-001 Rev A
- Page 7

RUCKUS Product Documentation Resources

Visit the RUCKUS website to locate related documentation for your product and additional RUCKUS resources.

Release Notes and other user documentation are available at <https://support.ruckuswireless.com/documents>. You can locate the documentation by product or perform a text search. Access to Release Notes requires an active support contract and a RUCKUS Support Portal user account. Other technical documentation content is available without logging in to the RUCKUS Support Portal.

White papers, data sheets, and other product documentation are available at <https://www.commscope.com/ruckus>.

Online Training Resources

To access a variety of online RUCKUS training modules, including free introductory courses to wireless networking essentials, site surveys, and products, visit the RUCKUS Training Portal at <https://commscopeuniversity.myabsorb.com/>. The registration is a two-step process described in this [video](#). You create a CommScope account and then register for, and request access for, CommScope University.

Document Conventions

The following table lists the text conventions that are used throughout this guide.

TABLE 1 Text Conventions

Convention	Description	Example
monospace	Identifies command syntax examples	<code>device(config)# interface ethernet 1/1/6</code>
bold	User interface (UI) components such as screen or page names, keyboard keys, software buttons, and field names	On the Start menu, click All Programs .
<i>italics</i>	Publication titles	Refer to the <i>RUCKUS Small Cell Release Notes</i> for more information.

Notes, Cautions, and Safety Warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE

A NOTE provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION

An ATTENTION statement indicates some information that you must read before continuing with the current action or task.



CAUTION

A CAUTION statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.



DANGER

A DANGER statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Command Syntax Conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
<i>italic text</i>	Identifies a variable.
[]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{x y z}	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x y	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, <i>member[member...]</i> .
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

About This Document

- [New in This Document](#) 9
- [Supported Hardware](#)..... 9

New in This Document

The following tables describe information added or modified in this guide for the FastIron 10.0.00 software release.

TABLE 2 Summary of Enhancements in FastIron Release 10.0.00

Feature	Description	Described in
RUCKUS ICX 8200	New: Added support for RUCKUS ICX 8200 devices.	Throughout the guide
Discontinuation of support for a separate switch image	Updated: This release and future releases support a single image for ICX platforms that provides both switching and routing functionality. As a result, support for a separate switch ("Layer 2") image is discontinued.	References have been removed throughout the guide
RESTCONF support for configuration archive.	This feature integrates configuration archive with RESTCONF, which uses structured data.	Refer to Configuring ICX Using the RESTCONF Management Interface on page 21 for more information.
Unsupported hardware	Updated: This release and future releases do not support RUCKUS ICX 7150, 7250, or 7450 devices.	References have been removed throughout the guide.
Updates to address defects	Updated: Minor updates on content throughout to address defects.	All chapters
Minor editorial updates	Updated: Minor editorial updates were made throughout the guide.	All chapters

Supported Hardware

This guide supports the following RUCKUS products:

- RUCKUS ICX 8200 Switches
- RUCKUS ICX 7850 Switches
- RUCKUS ICX 7650 Switches
- RUCKUS ICX 7550 Switches

For information about what models and modules these devices support, refer to the hardware installation guide for the specific product family.

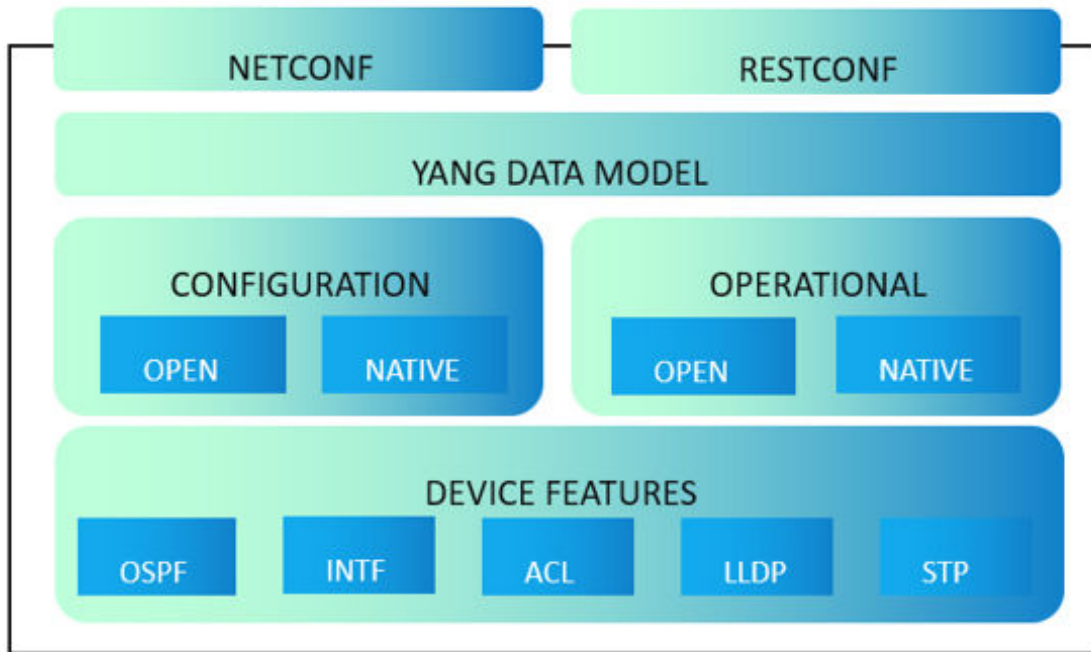
RESTCONF for RUCKUS ICX Switches

- Introduction..... 11
- RESTCONF-Supported HTTP Methods..... 13
- RESTCONF Supported Modules..... 13
- ICX Management..... 14
- RESTCONF Resources..... 16
- XML Representation..... 18
- JSON Representation..... 18
- Media Types..... 19
- Capabilities..... 19
- Schema Resources..... 19
- YANG Module Retrieval..... 20
- Query Parameters..... 20
- Transport Protocol Requirements..... 20
- Web Requests with Tools (curl)..... 21
- RESTCONF Configuration..... 21

Introduction

The RUCKUS ICX platform supports the RESTCONF protocol as defined in IETF RFC 8040. RESTCONF is a Hypertext Transfer Protocol (HTTP)-based protocol that uses Transport Layer Security (TLS) and provides a programmatic interface for accessing data defined in YANG, using the datastore concepts defined in the Network Configuration Protocol (NETCONF). RESTCONF is an IETF standard protocol, which is a successor of RESTful APIs and NETCONF and is used to configure and retrieve operational data on ICX. It is driven by YANG models that describe the configuration and operation of the ICX platform including validation rules and dependencies.

FIGURE 1 RESTCONF Protocol Stack



The following components make up the RESTCONF protocol stack:

- NETCONF
- YANG
- REST
- RESTCONF

NETCONF: Network Configuration Protocol (NETCONF) is a protocol used to manage networking devices like switches and routers. NETCONF uses Remote procedure calls (RPCs) to read, write configuration data into networking devices. Remote procedure calls use XML to send and receive configuration data from NETCONF management station to networking devices.

YANG: Yet Another Next Generation (YANG) is a data modeling language which can be used to send, receive and store data with network management protocols like RESTCONF and NETCONF. Yang is used to store both configuration and state of network elements.

REST: Representational state transfer (REST) is an API (Application programmer interface) used by web services. REST is a stateless client/server protocol uses http methods for requests between clients and servers. REST uses http methods for requests and responses with XLM or JASON for data encoding.

A Web server which uses REST APIs for client/server requests and responses is called RESTful.

RESTCONF: Is similar to REST API which is defined in RFC 8040. It is an HTTP-based protocol that provides a programmatic interface for accessing data defined in YANG, using the datastore concepts defined in NETCONF. As laid out in RFC 8040, "RESTCONF is an HTTP-based protocol that provides a programmatic interface for accessing data defined in YANG, using the datastore concepts defined in NETCONF."

Because RESTCONF uses an HTTP-based protocol to send and receive messages for networking devices, it has become a better option than NETCONF.

HTTP: Methods like GET, POST, PUT, DELETE, etc are used by REST/RESTCONF to send requests/responses. Organizations can use REST APIs to manage web services and RESTCONF to manage networking devices. Both use HTTP-based protocols, making it easy for operations and maintenance.

An external client can use XML or JSON YANG-based configurations to program the ICX platform configuration. The validation rules and dependencies described in the YANG model are useful to check user input even before sending the configuration request to the ICX platform. The data exchange format between clients and the ICX platform is JSON over a TLS/HTTP or HTTPS connection.

The OpenConfig working group has defined a common set of models to configure and operate network devices. The ICX utilizes the OpenConfig YANG model for its configuration and operation. The IETF has several YANG models which may be used when there is no model for a specific feature. An ICX custom model will be created if no standard-based models exist. YANG models will be deviated from and augmented to fit the ICX features and behaviors.

Refer to <http://www.openconfig.net/projects/models/> for more information on the OpenConfig working group.

RESTCONF-Supported HTTP Methods

The following configuration operations are supported:

TABLE 3 RESTCONF-Supported HTTP Methods

HTTP Method	Operation	Media Type
POST	Create (config)	application/yang.data
PUT	Replace (config)	application/yang.data
PATCH	Merge (config)	application/yang.data
DELETE	Delete (config)	application/yang.data
POST	Operation	application/yang.operation
GET	Read (config and operational)	application/yang.data
OPTIONS		application/yang.data
HEAD	Header metadata	No response body

Actions and operations are invoked using POST operations (for example, reboot).

RESTCONF Supported Modules

The following features are targeted for this release, only a subset of the equivalent CLIs for each feature will be supported.

NOTE

For the initial release of RESTCONF for ICX these are the features supported and even for these features not all functionality is supported.

ICX Feature	Openconfig Model
ICX Feature	Openconfig Model
Interface/Port	Config/Operational state
Vlan	Config
VE	Config
LAG	Config
POE	Config

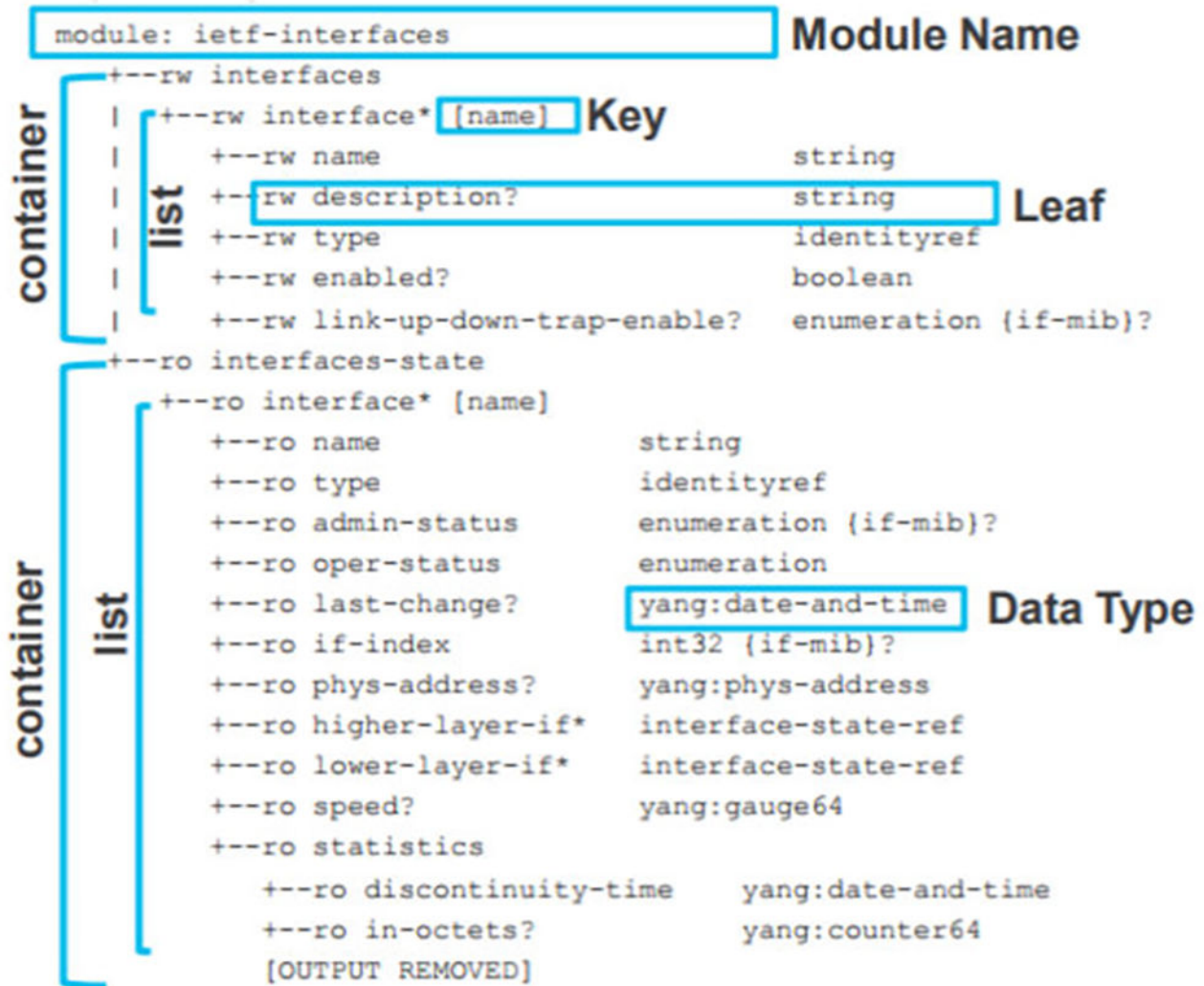
ICX Feature	Openconfig Model
LLDP	Config
Static Route	Config
OSPF	Config
ACL	Config
AAA	Config
Username / Passwords	Config
DNS	Config
STP	Config

ICX Management

RUCKUS ICX supports Web management and RESTCONF interfaces. The HTTPS Web service and RESTCONF use default TCP port 443. RESTCONF supports only HTTPS requests. RESTCONF requests are authenticated with a username and password (mandatory) and using SSL certificates (optional).

The following components make up the YANG model:

FIGURE 2 YANG Components



- Container: A collection of information logically grouped, such as a container for configuration and a container for state.
- List: Within a container you can have a list or even multiple lists, such as a list of interfaces.
- Key: Each item within the list is referenced with a key.
- Leaf: Within the list, a list contains the information.
- Data Type: Each leaf is associated with at data type.

RESTCONF Resources

URI

The Uniform Resource Identifier (URI) identifies the resource. The resources are represented with URIs in the following format:

`/restconf/<path>? <query>`

- `restconf`: The entry point of the URI in the device, and the root of the API configured on the device is discovered by getting the `"/.well-known/host-meta"` resource.
- `path`: The target resource URI, which is used for identifying the resource being accessed by the HTTP operation.
- `query`: A query parameter lists with the form of "name=value" pairs. Most query parameters (such as depth) are optional to implement by the server and optional to use by the client. Any reserved characters must be percent-encoded, according to RFC 3986. Refer to the supported query parameters at <https://tools.ietf.org/html/rfc8040#section-4.8>.

Base URI

The API resource contains the RESTCONF root resource for the RESTCONF datastore and operation resources. It is the top-level resource located at `{+restconf}` and has the media type "application/yang-data+xml" or "application/yang-data+json".

YANG tree diagram for an API resource:

```
+---- {+restconf}
+---- data
| ...
+---- operations?
| ...
+--ro yang-library-version string
```

The base URI for the RESTCONF API is : `/restconf/`.

The base URI `/restconf` contains the child resources below. The YANG representation is:

```
$ curl https://10.176.156.34/restconf --insecure -u testuser:testpassword -H "Accept: application/xml" |
xmllint --format - % Total % Received % Xferd
Average Speed Time Time Time Current Dload Upload Total Spent Left Speed 100 323 0 323 0 0 2323 0 --:--:--
--:--:-- --:--:-- 2323 2016-06-21 $ curl
https://10.176.156.34/restconf --insecure -u testuser:testpassword | json_pp % Total % Received % Xferd
Average Speed Time Time Time Current Dload
Upload Total Spent Left Speed 100 149 0 149 0 0 1006 0 --:--:-- --:--:-- --:--:-- 1000 { "ietf-
restconf:restconf" : { "yumaworks-restconf:yang" : [ null ], "data" :
{ }, "yang-library-version" : "2016-06-21", "operations" : { "yuma-system:no-op" : [ null ] } } }
```

Data

Data is a mandatory resource representing the combined configuration and state data resources that can be accessed by a client. It cannot be created or deleted by the client. The `"{+restconf}/data"` subtree represents the datastore resource, which is a collection of configuration data and state data nodes.

YANG Library Version

The "yang-library-version" is a mandatory leaf identifying the revision date of the "ietf-yang-library" YANG module that is implemented by this server.

Operation Resource

The operation resource represents an Remote Procedure Call (RPC) operation defined with the YANG "rpc" statement or a data-model-specific action defined with a YANG "action" statement. The statement is invoked using a POST method on the operation resource. You can use the GET Method on the /restconf/operations to check the list of RPCs it supports.

CRUD Operations

The RESTCONF protocol uses HTTP methods to identify the CRUD operations requested for a particular resource. The HTTP methods are used for manipulating the resource defined in the YANG model for the create, read, update, and delete (CRUD) operations.

POST Method

The RESTCONF POST method represents the NETCONF "create" operation. The POST method is sent by the client to create a data resource or invoke an operation resource. The server uses the target resource type to determine how to process the request.

Both the POST and PUT methods can be used to create data resources. The difference is that for POST, the client does not provide the resource identifier for the resource that will be created. The target resource for the POST method for resource creation is the parent of the new resource. The target resource for the PUT method for resource creation is the new resource itself.

TABLE 4 Resource Types Supporting POST

Type	Description
Datastore	Creates a top-level configuration data resource
Data	Creates a configuration data child resource
Operation	Invokes an RPC operation

PUT Method

The RESTCONF PUT method represents the NETCONF "create" or "replace" operation.

NOTE

A request message representing the new data resource must be present or the server will return a "400 Bad Request" message. The error tag value "invalid-value" is used in this case.

The PUT method on the datastore resource is used to replace the entire contents of the datastore. The PUT method on a data resource only replaces that data resource within the datastore.

TABLE 5 Resource Types Supporting PUT

Type	Description
Datastore	Replaces the entire contents of the datastore
Data	Replaces that data resource within the datastore

The "insert" and "point" query parameters are supported by the PUT method for data resources. These parameters are only allowed if the list or leaf-list is "ordered-by user". The query parameters are not allowed for the requests for a datastore resource.

- If the PUT request creates a new resource, a "201 Created" message is returned. If an existing resource is modified, a "204 No Content" message is returned.
- If the user is not authorized to create or replace the target resource, a "403 Forbidden" message is returned. The error tag value "access-denied" is used in this case.

RESTCONF for RUCKUS ICX Switches

XML Representation

- If the target resource represents a YANG leaf-list, then the PUT method *must not* change the value of the leaf-list instance.
- If the target resource represents a YANG list instance, then the key leaf values in the message-body representation *must* be the same as the key leaf values in the request URI. The PUT method *must not* be used to change the key leaf values for a data resource instance.

PATCH Method

RESTCONF uses the HTTP PATCH method defined to provide an extensible framework for resource-patching mechanisms. Each patch mechanism requires a unique media type.

For a leaf-list case, the PATCH method *must not* be used to change the key values of the leaf-list instance.

- If the target resource instance does not exist, the server *must not* create it.
- If the PATCH request succeeds, a "200 OK" message is returned if there is a message-body. The "204 No Content" message is returned if no response message-body is sent.
- If the user authentication fails, the "401 Unauthorized" message is returned.
- If the user is not authorized to alter the target resource, an error response containing a "403 Forbidden" status-line will be returned. A server may return a "404 Not Found" status-line. The error-tag value "invalid-value" is used in this case.

DELETE Method

RESTCONF uses the HTTP DELETE method defined to delete a resource if it exists. If the resource does not exist, the "404 Not Found" message is returned. The corresponding error tag value "data-missing" is also returned.

If the DELETE requests succeed, a "204 No Content" message is returned. In the case of a user authentication a "401 Unauthorized" message is returned.

XML Representation

A resource is represented as an XML element which contains the values of the resource (if any) with child elements to represent the sub-resources. An XML representation of a resource is used in both the request payload and in the response. The "xmlns" XML attribute is mentioned in the representation. This attribute has the name of the YANG module of the resource specified in the representation.

The UTF-8 character set is used for the XML message encoding.

TABLE 6 XML Message Encoding

Special Character	Encoded Character	Description of the Special Character
<	<	less-than symbol
>	>	greater-than symbol
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

JSON Representation

The ICX platform supports JSON format to represent the resource. The YANG elements in the resource models are mapped into JSON elements for the proper serialization.

- A leaf element is mapped into a single key-value pair. The key and the value are separated by a colon.

- A container element is mapped into a JSON object. Therefore, the equivalent representation of a container starts with a left curly bracket and ends with a right curly bracket. The elements within the container are separated with a comma.
- A list element is mapped into a JSON array. Therefore, the equivalent representation of the list starts with a left square bracket and ends with a right square bracket. The instances of the list element are separated by a comma.

Media Types

Media types determine the type of data contained within a resource representation. There are two media types to identify the different kinds of resources. It is specified in the Accept and Content-Type header values for the request and the response respectively.

TABLE 7 Media Types

Media Type	Resource
application/yang-data+xml	Represents the data resource derived from a YANG module in the XML format.
application/yang-data+jsonl	Represents any data resource derived from a YANG module in the JSON format.

Capabilities

This mandatory container holds the RESTCONF protocol capability URIs supported by the server. The server may maintain a last-modified timestamp for this container and return the "Last-Modified" header field when this data node is retrieved with the GET or HEAD methods. Note that the last-modified timestamp for the datastore resource is not affected by changes to this subtree.

The server maintains an entity-tag for this container and returns the "ETag" header field when this data node is retrieved with the GET or HEAD methods. Note that the entity-tag for the datastore resource is not affected by changes to this subtree. The server must include a "capability" URI leaf-list entry for the "defaults" mode used by the server.

The server must include a "capability" URI leaf-list entry identifying each supported optional protocol feature. This entry includes optional query parameters and may include other capability URIs (not defined in this document).

FIGURE 3 Capabilities Example

```
- <capabilities>
  <capability>urn:ietf:params:restconf:capability:depth:1.0</capability>
  <capability>urn:ietf:params:restconf:capability:with-defaults:1.0</capability>
  <capability>urn:ietf:params:restconf:capability:defaults:1.0?basic-mode=explicit</capability>
  <capability>urn:ietf:params:restconf:capability:fields:1.0</capability>
  <capability>urn:ietf:params:restconf:capability:replay:1.0</capability>
  <capability>urn:ietf:params:restconf:capability:filter:1.0</capability>
  <capability>urn:ietf:params:restconf:capability:yang-patch:1.0</capability>
</capabilities>
```

Schema Resources

You can retrieve YANG modules using the URL shown in the schema. The leaf "schema" must be present in the associated "module" list entry. To retrieve a YANG module, you must first get the URL for retrieving the schema, which is stored in the "schema" leaf. The client can get the URL to retrieve the schema. When the client responds, the URL and the corresponding YANG module can be obtained.

```
<module xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
  <name>icx-openconfig-if-poe-aug</name>
  <revision>2020-01-04</revision>
  <schema>http://localhost/restconf/yang/icx-openconfig-if-poe-aug/2020-01-04</schema>
  <namespace>http://commscope.com/ns/yang/icx/icx-openconfig-if-poe-aug</namespace>
```

```
<conformance-type>implement</conformance-type>
</module>
```

To download a specific YANG model, retrieve and module using the URL provided in the schema.

YANG Module Retrieval

The "ietf-yang-library" module provides the information about the YANG modules and submodules defined in the FastIron software. All YANG modules and submodules must be identified in the YANG module library.

- **modules:** This mandatory container holds the identifiers for the YANG data model modules supported by the server.
- **modules/module:** This mandatory list contains one entry for each YANG data model module supported by the server. There must be an instance of this list for every YANG module that is used by the server. This ietf-yang-library module is defined in the RFC 7895.

Query Parameters

Each RESTCONF operation allows zero or more query parameters to be present in the requested URI. The specific parameters that are allowed depend on the resource type, and sometimes the specific target resource used in the request.

Query parameters can be given in any order. Each parameter can appear once at most in a requested URI. A default value may apply if the parameter is missing.

TABLE 8 RESTCONF Query Parameters

Value	Methods	Description
content	GET	Used to select the types of data child resources (configuration and/or non-configuration) that are returned by the server.
depth	GET	Requests limited subtree depth in the reply content.
fields	GET	Request a subset of the target resource contents filter.
filter	GET	Boolean notification filter for event stream resources.
insert	POST/PUT	Insertion mode for user-ordered data resources.
point	POST/PUT	Insertion point for user-ordered data resources.
start-time	GET	Replay buffer start time for event stream resources.
stop-time	GET	Replay buffer stop time for event stream resources.
with-defaults	GET	Controls the retrieval of default values.

Transport Protocol Requirements

RESTCONF requires the following transport protocols:

- RESTCONF server is not supported over HTTP. To support data integrity and confidentiality, RESTCONF requires HTTPS.
- RESTCONF supports the "https" URI scheme, and ICX uses the IANA assigned default port 443.
- The RESTCONF server must authenticate client access to any protected resource. If the RESTCONF client is not authenticated, the server must send an HTTP response with "401 Authorization Required".

Web Requests with Tools (curl)

- To send a simple GET request, the following command can be used: `curl https://%3Cmgmt%20IP%3E/restconf/data/restconf-state/capabilities%20-H%20%22Accept:application/yang-data+xml%22%20--insecure%20-u%20%3Cusername%3E:%3Cpassword%3E`
- In order to retrieve data in different encoding use `--header (-H)` parameter with specific Accept header value (XML or JSON). The following request "Accepts" replies in JSON encoding: `curl https://%3Cmgmt%20IP%3E/restconf/data/restconf-state/capabilities%20-H%20%22Accept:application/yang-data+json%22%20--insecure%20-u%20%3Cusername%3E:%3Cpassword%3E`
- The same request with RESTCONF query parameters, for example, "depth" would look as follows: `curl https://%3Cmgmt%20IP%3E/restconf/data/restconf-state/capabilities?depth=1`

RESTCONF Configuration

Configuring ICX Using the RESTCONF Management Interface

```

HTTPS OPTIONS Request
curl -i -k -X "OPTIONS" https://<MGMT-IP>/restconf/data/lldp -H 'Accept: application/yang-data+json' -u
<username>:<password>
HTTP/1.1 204 No Content
Server: nginx/1.14.0
Date: Tue, 06 Jan 1970 10:10:35 GMT
Connection: keep-alive
Allow: OPTIONS,HEAD,GET,POST,PUT,PATCH,DELETE  □ Allowed methods on this URL
Accept-Patch: application/yang-patch

```

```

HTTPS HEAD Request
$ curl -i -k -I https://<MGMT-IP>/restconf/data/lldp -u <username>:<password>
HTTP/1.1 200 OK
Server: nginx/1.14.0
Date: Tue, 06 Jan 1970 10:38:08 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Cache-Control: no-cache
Pragma: no-cache

```

To create a new entry using the PUT method and an input file, use the following PUT request.

```

HTTPS PUT Request (Create or Replace Configuration)
curl -vX PUT -d @input.json --header "Content-Type: application/yang-data+json" https://<MGMT-IP>/restconf/
data/network-instances/network-instance/default-vrf/vlans/vlan=200 --insecure -u
<username>:<password>
Where input.json is:
{
  "vlan": [
    {
      "vlan-id": "200",
      "config": {
        "vlan-id": "200",
        "name": "VLAN 200"
      }
    }
  ]
}
$ curl -i -k -X "PUT" https://<MGMT-IP>/restconf/data/network-instances/network-instance/default-vrf/vlans/
vlan=200 -H 'Content-Type: application/yang-data+json' -H 'Accept:
application/yang-data+json' -u <username>:<password> -d @"input.json"
HTTP/1.1 201 Created
Server: nginx/1.14.0
Date: Tue, 06 Jan 1970 10:34:10 GMT
Transfer-Encoding: chunked
Connection: keep-alive

```

RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

```
Location: http://localhost/restconf/data/network-instances/network-instance/default-vrf/vlans/vlan=200
Cache-Control: no-cache
Pragma: no-cache
Last-Modified: Tue, 06 Jan 1970 10:34:10 GMT
ETag: 258
```

To create a new entry using the PATCH method and an input file, use the following PATCH request.

HTTPS PATCH Request (Merge Resource Configuration)

```
curl -vX PATCH https://<MGMT-IP>/restconf/data/network-instances/network-instance/default-vrf/vlans -d
@input.json --header "Content-Type:
application/yang-data+json" --insecure -u <username>:<password>
Where, input.json is
```

```
{
  "vlans" : {
    "vlan": [
      {
        "vlan-id": "100",
        "config": {
          "vlan-id":"100",
          "name":"VLAN 100"
        }
      }
    ]
  }
}
```

```
$ curl -i -k -X "PATCH" https://<MGMT-IP>/restconf/data/network-instances/network-instance/default-vrf/
vlans -H 'Content-Type: application/
yang-data+json' -H 'Accept: application/yang-data+json' -u <username>:<password> -d @"vlan_patch.json"
HTTP/1.1 204 No Content
Server: nginx/1.14.0
Date: Tue, 06 Jan 1970 10:31:09 GMT
Connection: keep-alive
Cache-Control: no-cache
Pragma: no-cache
Last-Modified: Tue, 06 Jan 1970 10:31:09 GMT
ETag: 253
```

To create a new entry using the POST method and an input file, use the following POST request.

HTTPS POST Request (Create resource)

```
curl -vX POST https://<MGMT-IP>/restconf/data/network-instances/network-instance/default-vrf/protocols/
protocol/STATIC,icx-static/static-routes -d @input.json --
header "Content-Type: application/yang-data+json" --insecure -u <username>:<password>
Where, input.json is
```

```
{
  "static": [
    {
      "prefix": "55.55.55.55/32",
      "config": {
        "prefix": "55.55.55.55/32"
      }
    }
  ]
}
```

```
$ curl -i -k -X "POST" https://<MGMT-IP>/restconf/data/network-instances/network-instance/default-vrf/
protocols/protocol/STATIC,icx-static/static-routes -H
'Content-Type: application/yang-data+json' -H 'Accept: application/yang-data+json' -u <username>:<password>
-d @"static_route_v0.json"
HTTP/1.1 201 Created
Server: nginx/1.14.0
Date: Tue, 06 Jan 1970 10:25:12 GMT
Transfer-Encoding: chunked
Connection: keep-alive
Location: http://localhost/restconf/data/network-instances/network-instance/default-vrf/protocols/protocol/
STATIC,icx-static/static-routes/static=55.55.55.55%2F32
Cache-Control: no-cache
Pragma: no-cache
Last-Modified: Tue, 06 Jan 1970 10:25:12 GMT
ETag: 250
```

To delete an existing configuration, use the following Delete request:

```
HTTPS Delete Request
curl -vX DELETE https://<MGMT-IP>/restconf/data/network-instances/network-instance/default-vrf/vlans/
vlan=100 --insecure -u <username>:<password>
```

- <https://<MGMT-IP>/restconf/data/ldp/config?with-defaults=report-all>
- <https://<MGMT-IP>/restconf/data/network-instances/network-instance/default-vrf/protocols/protocol/STATIC,icx-static/static-routes>
- <https://<MGMT-IP>/restconf/data/network-instances/network-instance/default-vrf/protocols/protocol/STATIC,icx-static?depth=3>
- <https://<MGMT-IP>/restconf/data/openconfig-interfaces:interfaces/interface=ethernet%201%2F1%2F1>
- [https://<MGMT-IP>/restconf/data?fields=ietf-yang-library:modules-state/module\(name;revision\)](https://<MGMT-IP>/restconf/data?fields=ietf-yang-library:modules-state/module(name;revision))
- <https://<MGMT-IP>/restconf/yang/icx-openconfig-aaa-aug/2019-09-04>

Archive File Upload and Download Operations

The URI to download an archive file is:

```
data/icx-openconfig-archive:archive-top/management/xfer/archive-download
```

The URI to upload an archive file is:

```
data/icx-openconfig-archive:archive-top/management/xfer/archive-upload
```

Supported HTTP Operations

POST method

```
/data/icx-openconfig-archive:archive-top/management/xfer/archive-download
```

```
Request body:
{"icx-openconfig-archive:input":
{"t_type": "https",
"ip-address" : "xx.xx.xx.xx",
"port": "443",
"archive-name": "ICX7K_ARCHIVE_00",
"filename": "ext_ICX7K_ARCHIVE_00"}}
```

Note: Port is optional

```
Response body: If successful,HTTP/1.1 204 No Content
If there is an error,HTTP/1.1 500 Internal Server Error
```

The archive name "ICX7K_ARCHIVE_00" does not exist, but the filename "ext_ICX7K_ARCHIVE_00" must exist.

POST method

```
/data/icx-openconfig-archive:archive-top/management/xfer/archive-download
```

```
Request body:
{"icx-openconfig-archive:input":
{"t_type": "tftp",
"ip-address" : "xx.xx.xx.xx",
"port": "69",
"archive-name": "ICX7K_ARCHIVE_00",
"filename": "ext_ICX7K_ARCHIVE_00"}}
```

Note: Port is optional

```
Response body: If successful,HTTP/1.1 204 No Content
If there is an error,HTTP/1.1 500 Internal Server Error
```

The archive name "ICX7K_ARCHIVE_00" does not exist, but the filename "ext_ICX7K_ARCHIVE_00" must exist.

RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

POST method

```
/data/icx-openconfig-archive:archive-top/management/xfer/archive-upload
```

Request body:

```
"icx-openconfig-archive:input":  
{  
  "t_type": "https",  
  "ip-address" : "xx.xx.xx.xx",  
  "port": "443",  
  "archive-name": "ICX7K_ARCHIVE_00",  
  "filename": "ext_ICX7K_ARCHIVE_00"}}}
```

Note: Port is optional

Response body: If successful, HTTP/1.1 204 No Content
If there is an error, HTTP/1.1 500 Internal Server Error

The archive name "ICX7K_ARCHIVE_00" must exist. The filename "ext_ICX7K_ARCHIVE_00" may cause the upload failure like "write protection error", depending on the server.

POST method

```
/data/icx-openconfig-archive:archive-top/management/xfer/archive-upload
```

Request body:

```
{"icx-openconfig-archive:input":  
{  
  "t_type": "tftp",  
  "ip-address" : "10.176.195.253",  
  "port": "69",  
  "archive-name": "ICX7K_ARCHIVE_00",  
  "filename": "ext_ICX7K_ARCHIVE_00"}}}
```

Note: Port is optional

Response body: If successful, HTTP/1.1 204 No Content
If there is an error, HTTP/1.1 500 Internal Server Error

The archive name "ICX7K_ARCHIVE_00" must exist. The filename "ext_ICX7K_ARCHIVE_00" may cause an upload failure such as "write protection error", depending on the server.

Listing of Archive Files

The URI to list all archives in a repository is:

```
/data/icx-openconfig-archive:archive-top/management/act-list
```

Supported HTTP Operations

POST method

```
/data/icx-openconfig-archive:archive-top/management/act-list
```

Request body:

```
{"icx-openconfig-archive:input":  
{"order-type": "ascending-order"}}
```

```
Response body: {"icx-openconfig-archive:output":{"list-result":"current_archive_files=4  
[maximum number of archives allowed=100]\n\nList of archives:\nICX7K_ARCHIVE_11 135 Mar 28  
09:04\nICX7K_ARCHIVE_10      3203 Mar 27  
09:35\nICX7K_ARCHIVE_01      3165 Mar 27  
07:16\nICX7K_ARCHIVE_00      3165 Mar 27  
07:15\n\n"}}}
```

Error: HTTP/1.1 500 Internal Server Error

POST method

```
/data/icx-openconfig-archive:archive-top/management/act-list
```

Request body:

```
{"icx-openconfig-archive:input":  
{"order-type": "descending-order"}}
```

Response body:

```
{"icx-openconfig-archive:output":{"list-result":"current_archive_files=4  
[maximum number of archives allowed=100]\n\nList of archives:ICX7K_ARCHIVE_00 3165 Mar 27  
07:15\nICX7K_ARCHIVE_01 3165 Mar 27  
07:16\nICX7K_ARCHIVE_10 3203 Mar 27  
09:35\nICX7K_ARCHIVE_11 135 Mar 28  
09:04\n\n"}}
```

Creating an Archive

The URI to create an archive is:

```
/data/icx-openconfig-archive:archive-top/management/act-create
```

Supported HTTP Operations

POST method

```
/data/icx-openconfig-archive:archive-top/management/act-create
```

Request body:

```
{"icx-openconfig-archive:input":  
{"creation-type": "rename-archive",  
"archive-name1": "ICX7K_ARCHIVE_00",  
"archive-name2": "ext_ICX7K_ARCHIVE_00"}}
```

Response body: If successful,HTTP/1.1 204 No Content
If there is an error,HTTP/1.1 500 Internal Server Error

The archive-name1 must exist on a target system and the archive-name2 does not exist.

POST method

```
/data/icx-openconfig-archive:archive-top/management/act-create
```

Request body:

```
{"icx-openconfig-archive:input":  
{"creation-type": "copy-archive",  
"archive-name1": "ICX7K_ARCHIVE_00",  
"archive-name2": "ext_ICX7K_ARCHIVE_10"}}
```

Response body: If successful,HTTP/1.1 204 No Content
If there is an error,HTTP/1.1 500 Internal Server Error

The archive-name1 must exist on a target system and the archive-name2 does not exist.

POST method

```
/data/icx-openconfig-archive:archive-top/management/act-create
```

Request body:{"icx-openconfig-archive:input":

```
{"creation-type": "copy-runcfg",  
"archive-name1": "ICX7K_ARCHIVE_30"}}
```

Response body: If successful,HTTP/1.1 204 No Content
If there is an error,HTTP/1.1 500 Internal Server Error

The archive-name1 must not exist on a target system.

Deleting an Archive

The URI to delete an archive is:

```
/data/icx-openconfig-archive:archive-top/management/act-delete
```

Supported HTTP Operations

POST method

```
/data/icx-openconfig-archive:archive-top/management/act-delete
```

Request body:

```
{"icx-openconfig-archive:input":  
{"delete-type": "delete-archive",  
"archive-name": "ICX7K_ARCHIVE_00"}}
```

Response body: If successful, HTTP/1.1 204 No Content
If there is an error, HTTP/1.1 500 Internal Server Error

The archive-name must exist on a target system.

POST method

```
/data/icx-openconfig-archive:archive-top/management/act-delete
```

Request body:

```
{"icx-openconfig-archive:input":  
{"delete-type": "delete-backup"}}
```

Response body: If successful, HTTP/1.1 204 No Content
If there is an error, HTTP/1.1 500 Internal Server Error

Differences between Archives

The URI to get differences between two archives, an archive or startup-config, an archive or running-config is:

```
/data/icx-openconfig-archive:archive-top/diff
```

Supported HTTP Operations

POST method

```
/data/icx-openconfig-archive:archive-top/diff
```

Request body:

```
{"icx-openconfig-archive:input":  
{"order-type": "diff-archives",  
"archive-name1": "ICX7K_ARCHIVE_00",  
"archive-name2": "ICX7K_ARCHIVE_01"}}
```

Response body: If successful, HTTP/1.1 204 No Content
{"icx-openconfig-archive:output":
{"diff-result": "--- baseline\n+++ comparison\n\n1: logging console\n 2: vlan 10 by port\n\n"}}

If there is an error, HTTP/1.1 500 Internal Server Error

The two archives indicated by archive-name1 and archive-name2 must exist.

POST method

```
/data/icx-openconfig-archive:archive-top/diff
```

Request body:

```
{"icx-openconfig-archive:input":
```

```
{"order-type": "diff-archive-startup",  
"archive-name1": "ICX7K_ARCHIVE_00"}}
```

Response body: If successful,HTTP/1.1 204 No Content
If there is an error,HTTP/1.1 500 Internal Server Error

The archive name indicated by archive-name1 must exist.

POST method

```
/data/icx-openconfig-archive:archive-top/diff
```

Request body:
{"icx-openconfig-archive:input":
{"order-type": "diff-archive-runcfg",
"archive-name1": "ICX7K_ARCHIVE_00"}}

Response body: If successful,HTTP/1.1 204 No Content
If there is an error,HTTP/1.1 500 Internal Server Error

The archive name indicated by archive-name1 must exist.

Reloading an Archive

Supported HTTP Operations

POST method

```
/data/icx-openconfig-archive:archive-top/archive-reload
```

Request body:
{"icx-openconfig-archive:input":
{"partition-type": "primary",
"archive-name": "ICX7K_ARCHIVE_00"}}

Response body: None
If there is an error, HTTP/1.1 500 Internal Server Error

The archive-name must exist on a target system.

POST method

```
/data/icx-openconfig-archive:archive-top/archive-reload
```

Request body:
{"icx-openconfig-archive:input":
{"partition-type": "secondary",
"archive-name": "ICX7K_ARCHIVE_00"}}

Response body: None
If there is an error, HTTP/1.1 500 Internal Server Error

Archiving a Backup File

Supported HTTP Operations

POST method

```
/data/icx-openconfig-archive:archive-top/backup-download/download-action
```

Request body:
{"icx-openconfig-archive:input":
{"t_type": "tftps",
"ip-address" : "xx.xx.xx.xx",

RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

```
"port": "443",  
"filename": "file1"}}
```

Response body: If successful, HTTP/1.1 204 No Content
If there is an error, HTTP/1.1 500 Internal Server Error

POST method

```
/data/icx-openconfig-archive:archive-top/backup-download/download-action
```

```
Request body:  
{  
  "icx-openconfig-archive:input":  
  {  
    "t_type": "tftp",  
    "ip-address" : "xx.xx.xx.xx",  
    "port": "69",  
    "filename": "file1"}}
```

Response body: If successful, HTTP/1.1 204 No Content
If there is an error, HTTP/1.1 500 Internal Server Error

IPv4, IPv6 and MAC ACLs

Forwarding Reference is not supported in ACL. If ACL is not created for RESTCONF, then you cannot bind an ACL on an interface.

You cannot delete an ACL, if it is bind on any interface. For this, delete the ACL binding and the ACL for RESTCONF to delete an ACL on any physical interface.

ACL is supported on physical interface only. ACL is not supported on VLAN and LAG interfaces for RESTCONF.

If you want to update a filter of a particular sequence ID, first delete the filter and then add it back with modification.

Standard Named ACL is not possible.

Supported HTTP Operations for IPv4 ACL

PATCH method

To create or update IPv4 extended ACL using PATCH method.

```
URL: https://<host>/restconf/data/acl/acl-sets
```

```
{  
  "acl-sets": {  
    "acl-set": [  
      {  
        "name": "ext3",  
        "type": "ACL_IPV4",  
        "config": {  
          "name": "ext3",  
          "type": "ACL_IPV4"  
        },  
        "acl-entries": {  
          "acl-entry": [  
            {  
              "sequence-id": 10,  
              "config": {  
                "sequence-id": 10  
              },  
              "ipv4": {  
                "config": {  
                  "source-address": "0.0.0.0",  
                  "destination-address": "0.0.0.0",  
                  "dscp": 0,  
                  "protocol": 6  
                }  
              },  
              "transport": {
```

```

        "config": {
            "source-port": "0",
            "destination-port": "0"
        },
        "actions": {
            "config": {
                "forwarding-action": "openconfig-acl:ACCEPT"
            }
        }
    ]
}

```

POST method

To create IPv4 ACL with ACL entries using POST method.

URL: <https://<host>/restconf/data/acl/acl-sets>

```

{
  "acl-set": [
    {
      "name": "ext_acl1",
      "type": "ACL_IPV4",
      "config": {
        "name": "ext_acl1",
        "type": "ACL_IPV4"
      },
      "acl-entries": {
        "acl-entry": [
          {
            "sequence-id": 10,
            "config": {
              "sequence-id": 10
            },
            "ipv4": {
              "config": {
                "source-address": "0.0.0.0",
                "destination-address": "0.0.0.0",
                "dscp": 0,
                "protocol": 6
              }
            },
            "transport": {
              "config": {
                "source-port": "0",
                "destination-port": "0"
              }
            },
            "actions": {
              "config": {
                "forwarding-action": "openconfig-acl:ACCEPT"
              }
            }
          }
        ]
      }
    }
  ]
}

```

POST method

RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

To apply IPv4 ACL on ingress or egress ACL using POST method.

URL: `https://<host>/restconf/data/acl`

```
{
  "interfaces": {
    "interface": [
      {
        "id": "ethernet 1/1/1",
        "config": {
          "id": "ethernet 1/1/1"
        },
        "ingress-acl-sets": {
          "ingress-acl-set": [
            {
              "set-name": "acl1",
              "type": "ACL_IPV4",
              "config": {
                "set-name": "acl1",
                "type": "ACL_IPV4"
              }
            }
          ]
        }
      },
      {
        "id": "ethernet 1/1/2",
        "config": {
          "id": "ethernet 1/1/2"
        },
        "egress-acl-sets": {
          "egress-acl-set": [
            {
              "set-name": "acl1",
              "type": "ACL_IPV4",
              "config": {
                "set-name": "acl1",
                "type": "ACL_IPV4"
              }
            }
          ]
        }
      }
    ]
  }
}
```

PATCH method

To apply IPv4 ACL on ingress and egress interface using PATCH method.

URL: `https://<host>/restconf/data/acl/interfaces`

```
{
  "interfaces": {
    "interface": [
      {
        "id": "ethernet 1/1/1",
        "config": {
          "id": "ethernet 1/1/1"
        },
        "ingress-acl-sets": {
          "ingress-acl-set": [
            {
              "set-name": "acl1",
              "type": "ACL_IPV4",
              "config": {
                "set-name": "acl1",
                "type": "ACL_IPV4"
              }
            }
          ]
        }
      }
    ]
  }
}
```


RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

```
    "transport": {
      "config": {
        "source-port": "10",
        "destination-port": "10"
      },
      "state": {}
    },
    "input-interface": {
      "state": {},
      "interface-ref": {
        "state": {}
      }
    },
    "actions": {
      "config": {
        "forwarding-action": "openconfig-acl:DROP"
      },
      "state": {}
    }
  },
  {
    "sequence-id": 20,
    "config": {
      "sequence-id": 20
    },
    "state": {},
    "ipv4": {
      "config": {
        "source-address": "0.0.0.0",
        "destination-address": "0.0.0.0",
        "dscp": 0,
        "protocol": 0
      },
      "state": {}
    },
    "transport": {
      "config": {
        "source-port": "0",
        "destination-port": "0"
      },
      "state": {}
    },
    "input-interface": {
      "state": {},
      "interface-ref": {
        "state": {}
      }
    },
    "actions": {
      "config": {
        "forwarding-action": "openconfig-acl:ACCEPT"
      },
      "state": {}
    }
  }
]
}
]
```

GET method : To get or read specific ACL binded interface.

URL: <https://<host>/restconf/data/acl/interfaces/interface/ethernet 1%2F1%2F5>

```
{
  "openconfig-acl:interface": [
    {
      "id": "ethernet 1/1/5",
      "config": {
        "id": "ethernet 1/1/5"
      }
    }
  ]
}
```



```

    },
    "state": {},
    "interface-ref": {
      "config": {
        "interface": "ethernet 1/1/5"
      },
      "state": {}
    },
  },
  "ingress-acl-sets": {
    "ingress-acl-set": [
      {
        "set-name": "ip_stan_test",
        "type": "openconfig-acl:ACL_IPV4",
        "config": {
          "set-name": "ip_stan_test",
          "type": "openconfig-acl:ACL_IPV4"
        },
        "state": {},
        "acl-entries": {}
      }
    ]
  },
  "egress-acl-sets": {}
}
]

```

Supported HTTP operations for MAC ACL

To create or update MAC ACL using PATCH method.

URL: <https://<host>/restconf/data/acl/acl-sets>

```

{
  "acl-sets": {
    "acl-set": [
      {
        "name": "bj5",
        "type": "ACL_L2",
        "config": {
          "name": "bj5",
          "type": "ACL_L2"
        },
        "acl-entries": {
          "acl-entry": [
            {
              "sequence-id": 100,
              "config": {
                "sequence-id": 100
              },
              "l2": {
                "config": {
                  "source-mac": "1111.2222.3333",
                  "source-mac-mask": "1111.2222.3333",
                  "destination-mac": "1111.2222.3333",
                  "destination-mac-mask": "1111.2222.3333",
                  "ethertype": 2048
                }
              },
              "actions": {
                "config": {
                  "forwarding-action": "ACCEPT"
                }
              }
            }
          ]
        }
      }
    ]
  }
}

```

RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

To create MAC ACL using POST method.

URL: `https://<host>/restconf/data/acl/acl-sets`

```
{
  "acl-set": [
    {
      "name": "bj5",
      "type": "ACL_L2",
      "config": {
        "name": "bj5",
        "type": "ACL_L2"
      },
      "acl-entries": {
        "acl-entry": [
          {
            "sequence-id": 100,
            "config": {
              "sequence-id": 100
            },
            "l2": {
              "config": {
                "source-mac": "1111.2222.3333",
                "source-mac-mask": "1111.2222.3333",
                "destination-mac": "1111.2222.3333",
                "destination-mac-mask": "1111.2222.3333",
                "ethertype": 2048
              }
            },
            "actions": {
              "config": {
                "forwarding-action": "ACCEPT"
              }
            }
          }
        ]
      }
    }
  ]
}
```

To apply MAC ACL (L2 ACL) on ingress using POST method.

URL: `https://10.176.128.136/restconf/data/acl`

```
{
  "interfaces": {
    "interface": [
      {
        "id": "ethernet 2/1/19",
        "config": {
          "id": "ethernet 2/1/19"
        },
        "ingress-acl-sets": {
          "ingress-acl-set": [
            {
              "set-name": "bj5",
              "type": "ACL_L2",
              "config": {
                "set-name": "bj5",
                "type": "ACL_L2"
              }
            }
          ]
        }
      }
    ]
  }
}
```

To apply MAC ACL on ingress interface using PATCH.

URL: <https://<host>/restconf/data/acl/interfaces>

```
{
  "interfaces": {
    "interface": [
      {
        "id": "ethernet 1/1/10",
        "config": {
          "id": "ethernet 1/1/10"
        },
        "ingress-acl-sets": {
          "ingress-acl-set": [
            {
              "set-name": "bj5",
              "type": "ACL_L2",
              "config": {
                "set-name": "bj5",
                "type": "ACL_L2"
              }
            }
          ]
        }
      }
    ]
  }
}
```

To DELETE a specific MAC ACL (L2 ACL) configured in the system

URL: https://<host>/restconf/data/acl/acl-sets/acl-set/bj5/ACL_L2

To DELETE L2 ACL binding on ingress interface

URL: [https://<host>/restconf/data/acl/interfaces/interface/ethernet 1%2F1%2F10/ingress-acl-sets/ingress-acl-set/bj5/ACL_L2](https://<host>/restconf/data/acl/interfaces/interface/ethernet%201%2F1%2F10/ingress-acl-sets/ingress-acl-set/bj5/ACL_L2)

To GET or READ specific L2 ACL (MAC ACL) configured in the system

URL: https://<host>/restconf/data/acl/acl-sets/acl-set/bj5/ACL_L2

```
{
  "openconfig-acl:acl-set": [
    {
      "name": "bj5",
      "type": "openconfig-acl:ACL_L2",
      "config": {
        "name": "bj5",
        "type": "openconfig-acl:ACL_L2"
      },
      "state": {},
      "acl-entries": {
        "acl-entry": [
          {
            "sequence-id": 100,
            "config": {
              "sequence-id": 100
            },
            "state": {},
            "l2": {
              "config": {
                "source-mac": "1111.2222.3333",
                "source-mac-mask": "1111.2222.3333",
                "destination-mac": "1111.2222.3333",
                "destination-mac-mask": "1111.2222.3333",
                "ethertype": 2048
              },
              "state": {}
            }
          }
        ]
      }
    }
  ]
}
```

RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

```
        "input-interface": {
            "state": {},
            "interface-ref": {
                "state": {}
            }
        },
        "actions": {
            "config": {
                "forwarding-action": "openconfig-acl:ACCEPT"
            },
            "state": {}
        }
    }
}
]
```

To GET or READ specific MAC ACL binding.

URL: `https://<host?>/restconf/data/acl/interfaces/interface/ethernet 1%2F1%2F1`

```
{
  "openconfig-acl:interface": [
    {
      "id": "ethernet 1/1/1",
      "config": {
        "id": "ethernet 1/1/1"
      },
      "state": {},
      "interface-ref": {
        "config": {
          "interface": "ethernet 1/1/1"
        },
        "state": {}
      },
      "ingress-acl-sets": {
        "ingress-acl-set": [
          {
            "set-name": "bj6",
            "type": "openconfig-acl:ACL_L2",
            "config": {
              "set-name": "bj6",
              "type": "openconfig-acl:ACL_L2"
            },
            "state": {},
            "acl-entries": {}
          }
        ]
      },
      "egress-acl-sets": {}
    }
  ]
}
```

Supported HTTP Operations for IPv6 ACL

PATCH method : To create IPv6 ACL with ACL entries using PATCH method.

URL: `https://<host>/restconf/data/acl/acl-sets`

```
{
  "acl-sets": {
    "acl-set": [
      {
        "name": "acl_ipv61",
        "type": "ACL_IPV6",
        "config": {
          "name": "acl_ipv61",

```

```

        "type": "ACL_IPV6"
    },
    "acl-entries": {
        "acl-entry": [
            {
                "sequence-id": 10,
                "config": {
                    "sequence-id": 10
                },
                "ipv6": {
                    "config": {
                        "source-address": "1000::10/120",
                        "destination-address": "2000::10/100",
                        "dscp": 0
                    }
                },
                "actions": {
                    "config": {
                        "forwarding-action": "ACCEPT"
                    }
                }
            }
        ]
    }
}

```

POST method : To create IPv6 ACL with ACL entries using POST method.

URL: <https://<host>/restconf/data/acl/acl-sets>

```

{
  "acl-set": [
    {
      "name": "acl_ipv6",
      "type": "ACL_IPV6",
      "config": {
        "name": "acl_ipv6",
        "type": "ACL_IPV6"
      },
      "acl-entries": {
        "acl-entry": [
          {
            "sequence-id": 10,
            "config": {
              "sequence-id": 10
            },
            "ipv6": {
              "config": {
                "source-address": "1000::10/120",
                "destination-address": "2000::10/100",
                "dscp": 0,
                "protocol": "2"
              }
            },
            "actions": {
              "config": {
                "forwarding-action": "ACCEPT"
              }
            }
          }
        ]
      }
    }
  ]
}

```

RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

To apply or update IPv6 ACL on ingress and egress interface using PATCH method

URL: <https://<host>/restconf/data/acl/interfaces>

```
{
  "interfaces": {
    "interface": [
      {
        "id": "ethernet 1/1/3",
        "config": {
          "id": "ethernet 1/1/3"
        },
        "ingress-acl-sets": {
          "ingress-acl-set": [
            {
              "set-name": "acl_ipv6",
              "type": "ACL_IPV6",
              "config": {
                "set-name": "acl_ipv6",
                "type": "ACL_IPV6"
              }
            }
          ]
        }
      },
      {
        "id": "ethernet 1/1/4",
        "config": {
          "id": "ethernet 1/1/4"
        },
        "egress-acl-sets": {
          "egress-acl-set": [
            {
              "set-name": "acl_ipv6",
              "type": "ACL_IPV6",
              "config": {
                "set-name": "acl_ipv6",
                "type": "ACL_IPV6"
              }
            }
          ]
        }
      }
    ]
  }
}
```

To apply IPv6 ACL on ingress or egress interfaces using POST method.

URL: <https://<host>/restconf/data/acl>

```
{
  "interfaces": {
    "interface": [
      {
        "id": "ethernet 1/1/3",
        "config": {
          "id": "ethernet 1/1/3"
        },
        "ingress-acl-sets": {
          "ingress-acl-set": [
            {
              "set-name": "acl_ipv61",
              "type": "ACL_IPV6",
              "config": {
                "set-name": "acl_ipv61",
                "type": "ACL_IPV6"
              }
            }
          ]
        }
      }
    ]
  }
}
```

```

    },
    {
      "id": "ethernet 1/1/4",
      "config": {
        "id": "ethernet 1/1/4"
      },
      "egress-acl-sets": {
        "egress-acl-set": [
          {
            "set-name": "acl_ipv61",
            "type": "ACL_IPV6",
            "config": {
              "set-name": "acl_ipv61",
              "type": "ACL_IPV6"
            }
          }
        ]
      }
    }
  ]
}

```

To DELETE IPv6 ACL binding on ingress interface.

URL: https://<host>/restconf/data/acl/interfaces/interface/ethernet 1%2F1%2F3/ingress-acl-sets/ingress-acl-set/acl_ipv61/ACL_IPV6

To DELETE IPv6 ACL binding on egress interface.

URL: https://<host>/restconf/data/acl/interfaces/interface/ethernet 1%2F1%2F4/egress-acl-sets/egress-acl-set/acl_ipv61/ACL_IPV6

To DELETE a specific IPv6 ACL configured in the system

URL: https://<host>/restconf/data/acl/acl-sets/acl-set/acl_ipv61/ACL_IPV6

To GET or READ specific IPv6 ACL configured in the system

URL: https://<host>/restconf/data/acl/acl-sets/acl-set/acl_ipv61/ACL_IPV6

```

{
  "openconfig-acl:acl-set": [
    {
      "name": "acl_ipv61",
      "type": "openconfig-acl:ACL_IPV6",
      "config": {
        "name": "acl_ipv61",
        "type": "openconfig-acl:ACL_IPV6"
      },
      "state": {},
      "acl-entries": {
        "acl-entry": [
          {
            "sequence-id": 10,
            "config": {
              "sequence-id": 10
            },
            "state": {},
            "ipv6": {
              "config": {
                "source-address": "1000::10/120",
                "destination-address": "2000::10/100",
                "dscp": 0,
                "protocol": 6
              },
              "state": {}
            },
            "transport": {
              "config": {

```

RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

```
        "source-port": "0",
        "destination-port": "0"
    },
    "state": {}
},
"input-interface": {
    "state": {},
    "interface-ref": {
        "state": {}
    }
},
"actions": {
    "config": {
        "forwarding-action": "openconfig-acl:ACCEPT"
    },
    "state": {}
}
}
]
}
]
}
```

To GET or READ all ACL binding interfaces

URL: <https://<host>/restconf/data/acl/interfaces>

```
{
  "openconfig-acl:interfaces": {
    "interface": [
      {
        "id": "ethernet 1/1/10",
        "config": {
          "id": "ethernet 1/1/10"
        },
        "state": {},
        "interface-ref": {
          "config": {
            "interface": "ethernet 1/1/10"
          },
          "state": {}
        },
        "ingress-acl-sets": {
          "ingress-acl-set": [
            {
              "set-name": "ext_acl1",
              "type": "openconfig-acl:ACL_IPV4",
              "config": {
                "set-name": "ext_acl1",
                "type": "openconfig-acl:ACL_IPV4"
              },
              "state": {},
              "acl-entries": {}
            }
          ]
        },
        "egress-acl-sets": {}
      },
      {
        "id": "ethernet 1/1/12",
        "config": {
          "id": "ethernet 1/1/12"
        },
        "state": {},
        "interface-ref": {
          "config": {
            "interface": "ethernet 1/1/12"
          },
          "state": {}
        }
      }
    ]
  }
}
```



```

"ingress-acl-sets": {},
"egress-acl-sets": {
  "egress-acl-set": [
    {
      "set-name": "ext_acl1",
      "type": "openconfig-acl:ACL_IPV4",
      "config": {
        "set-name": "ext_acl1",
        "type": "openconfig-acl:ACL_IPV4"
      },
      "state": {},
      "acl-entries": {}
    }
  ]
},
{
  "id": "ethernet 1/1/3",
  "config": {
    "id": "ethernet 1/1/3"
  },
  "state": {},
  "interface-ref": {
    "config": {
      "interface": "ethernet 1/1/3"
    },
    "state": {}
  },
  "ingress-acl-sets": {
    "ingress-acl-set": [
      {
        "set-name": "acl_ipv61",
        "type": "openconfig-acl:ACL_IPV6",
        "config": {
          "set-name": "acl_ipv61",
          "type": "openconfig-acl:ACL_IPV6"
        },
        "state": {},
        "acl-entries": {}
      }
    ]
  },
  "egress-acl-sets": {}
},
{
  "id": "ethernet 1/1/4",
  "config": {
    "id": "ethernet 1/1/4"
  },
  "state": {},
  "interface-ref": {
    "config": {
      "interface": "ethernet 1/1/4"
    },
    "state": {}
  },
  "ingress-acl-sets": {},
  "egress-acl-sets": {
    "egress-acl-set": [
      {
        "set-name": "acl_ipv61",
        "type": "openconfig-acl:ACL_IPV6",
        "config": {
          "set-name": "acl_ipv61",
          "type": "openconfig-acl:ACL_IPV6"
        },
        "state": {},
        "acl-entries": {}
      }
    ]
  }
},
},

```

RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

```
{
  "id": "ethernet 1/1/5",
  "config": {
    "id": "ethernet 1/1/5"
  },
  "state": {},
  "interface-ref": {
    "config": {
      "interface": "ethernet 1/1/5"
    },
    "state": {}
  },
  "ingress-acl-sets": {
    "ingress-acl-set": [
      {
        "set-name": "ip_stan_test",
        "type": "openconfig-acl:ACL_IPV4",
        "config": {
          "set-name": "ip_stan_test",
          "type": "openconfig-acl:ACL_IPV4"
        },
        "state": {},
        "acl-entries": {}
      }
    ]
  },
  "egress-acl-sets": {}
},
{
  "id": "ethernet 1/1/6",
  "config": {
    "id": "ethernet 1/1/6"
  },
  "state": {},
  "interface-ref": {
    "config": {
      "interface": "ethernet 1/1/6"
    },
    "state": {}
  },
  "ingress-acl-sets": {},
  "egress-acl-sets": {
    "egress-acl-set": [
      {
        "set-name": "ip_stan_test",
        "type": "openconfig-acl:ACL_IPV4",
        "config": {
          "set-name": "ip_stan_test",
          "type": "openconfig-acl:ACL_IPV4"
        },
        "state": {},
        "acl-entries": {}
      }
    ]
  }
}
]
```

To READ or GET ACL configured in the system

URL: <https://<host>/restconf/data/acl/acl-sets>

```
{
  "openconfig-acl:acl-sets": {
    "acl-set": [
      {
        "name": "ext_acl1",
        "type": "openconfig-acl:ACL_IPV4",
        "config": {
```

```

        "name": "ext_acl1",
        "type": "openconfig-acl:ACL_IPV4"
    },
    "state": {},
    "acl-entries": {
        "acl-entry": [
            {
                "sequence-id": 10,
                "config": {
                    "sequence-id": 10
                },
                "state": {},
                "ipv4": {
                    "config": {
                        "source-address": "0.0.0.0",
                        "destination-address": "0.0.0.0",
                        "dscp": 0,
                        "protocol": 6
                    },
                    "state": {}
                },
                "transport": {
                    "config": {
                        "source-port": "0",
                        "destination-port": "0"
                    },
                    "state": {}
                },
                "input-interface": {
                    "state": {},
                    "interface-ref": {
                        "state": {}
                    }
                },
                "actions": {
                    "config": {
                        "forwarding-action": "openconfig-acl:ACCEPT"
                    },
                    "state": {}
                }
            }
        ]
    }
},
{
    "name": "acl_ipv61",
    "type": "openconfig-acl:ACL_IPV6",
    "config": {
        "name": "acl_ipv61",
        "type": "openconfig-acl:ACL_IPV6"
    },
    "state": {},
    "acl-entries": {
        "acl-entry": [
            {
                "sequence-id": 10,
                "config": {
                    "sequence-id": 10
                },
                "state": {},
                "ipv6": {
                    "config": {
                        "source-address": "1000::10/120",
                        "destination-address": "2000::10/100",
                        "dscp": 0,
                        "protocol": 6
                    },
                    "state": {}
                },
                "transport": {
                    "config": {
                        "source-port": "0",

```

```

        "destination-port": "0"
      },
      "state": {}
    },
    "input-interface": {
      "state": {},
      "interface-ref": {
        "state": {}
      }
    },
    "actions": {
      "config": {
        "forwarding-action": "openconfig-acl:ACCEPT"
      },
      "state": {}
    }
  }
]
},
{
  "name": "ip_stan_test",
  "type": "openconfig-acl:ACL_IPV4",
  "config": {
    "name": "ip_stan_test",
    "type": "openconfig-acl:ACL_IPV4"
  },
  "state": {},
  "acl-entries": {
    "acl-entry": [
      {
        "sequence-id": 10,
        "config": {
          "sequence-id": 10
        },
        "state": {},
        "ipv4": {
          "config": {
            "source-address": "10.157.29.12",
            "destination-address": "0.0.0.0",
            "dscp": 0,
            "protocol": 0
          },
          "state": {}
        },
        "transport": {
          "config": {
            "source-port": "0",
            "destination-port": "0"
          },
          "state": {}
        },
        "input-interface": {
          "state": {},
          "interface-ref": {
            "state": {}
          }
        },
        "actions": {
          "config": {
            "forwarding-action": "openconfig-acl:DROP"
          },
          "state": {}
        }
      },
      {
        "sequence-id": 20,
        "config": {
          "sequence-id": 20
        },
        "state": {},
        "ipv4": {

```

```

    "config": {
      "source-address": "0.0.0.0",
      "destination-address": "0.0.0.0",
      "dscp": 0,
      "protocol": 0
    },
    "state": {}
  },
  "transport": {
    "config": {
      "source-port": "0",
      "destination-port": "0"
    },
    "state": {}
  },
  "input-interface": {
    "state": {},
    "interface-ref": {
      "state": {}
    }
  },
  "actions": {
    "config": {
      "forwarding-action": "openconfig-acl:ACCEPT"
    },
    "state": {}
  }
}
]
},
{
  "name": "bj5",
  "type": "openconfig-acl:ACL_L2",
  "config": {
    "name": "bj5",
    "type": "openconfig-acl:ACL_L2"
  },
  "state": {},
  "acl-entries": {
    "acl-entry": [
      {
        "sequence-id": 100,
        "config": {
          "sequence-id": 100
        },
        "state": {},
        "12": {
          "config": {
            "source-mac": "1111.2222.3333",
            "source-mac-mask": "1111.2222.3333",
            "destination-mac": "1111.2222.3333",
            "destination-mac-mask": "1111.2222.3333",
            "ethertype": 2048
          },
          "state": {}
        },
        "input-interface": {
          "state": {},
          "interface-ref": {
            "state": {}
          }
        },
        "actions": {
          "config": {
            "forwarding-action": "openconfig-acl:ACCEPT"
          },
          "state": {}
        }
      }
    ]
  }
}
]
}

```

```
}  
  ]  
}
```

Enabling the RESTCONF Management Interface

Complete the following steps to enable the RESTCONF management interface on an ICX switch and specify the parameters for keeping the FastIron system configuration and RESTCONF configuration in sync.

Ensure the TPM/SSL certificates are already available on the device prior to enabling RESTCONF.

The RESTCONF protocol stack runs as a separate process, which is loaded with the YANG models and maintains its own configuration database with regards to the YANG data model. Therefore, the FastIron configuration must be synced to the RESTCONF protocol stack. The configuration synchronization is always on if RESTCONF is enabled.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable the RESTCONF management interface.

```
device(config)# restconf enable
```

3. (Optional) Enter the **config-sync** command to force a configuration synchronization from FastIron to RESTCONF.

```
device(config)# config-sync
```

4. (Optional) Enter the **enable-config-sync** command to enable configuration sync from FastIron to RESTCONF.

```
device(config)# enable-config-sync
```

5. (Optional) Enter the **config-sync-interval** command to set the periodic configuration sync timeout.

```
device(config)# enable-config-sync
```

6. (Optional) Enter the **show restconf config** command to display the RESTCONF configuration.

```
device(config)# show restconf config
```

The following commands can be run to see if the RESTCONF interface is running successfully:

- show log
- show hmon client status all

Displaying RESTCONF Configuration and Status Information

You can use the following show commands to display information about the RESTCONF configuration and status. All of the commands are optional and do not need to be entered in the specified order.

1. Enter the **show restconf config** command to show which RESTCONF parameters are configured.

```
device> show restconf config  
RESTConf Feature Configuration  
restconf enable  
protocol-debug-level error  
platform-debug-level info  
config-sync enabled  
config-sync-timeout 5 mins
```

2. Enter the **show restconf event-counters** command to display information about RESTCONF event counters.

```
device> show restconf event-counters
SCP Event |Processed |YANG Map |Queued |Duplicate |Unknown Oper |
153 |1 |1 |1 |0 |0 |
154 |0 |2 |0 |0 |0 |
155 |0 |3 |0 |0 |0 |
131 |2 |4 |1 |1 |0 |
143 |1 |5 |1 |0 |0 |
144 |0 |6 |0 |0 |0 |
161 |0 |9 |0 |0 |0 |
| |10 |0 |0 |0 |
140 |0 |8 |0 |0 |0 |
| |7 |0 |0 |0 |
141 |0 |11 |0 |0 |0 |
145 |0 |12 |0 |0 |0 |
146 |0 |13 |0 |0 |0 |
147 |0 |14 |0 |0 |0 |
148 |0 |15 |0 |0 |0 |
149 |0 |16 |0 |0 |0 |
156 |0 |17 |0 |0 |0 |
157 |0 |18 |0 |0 |0 |
158 |0 |19 |0 |0 |0 |
150 |0 |20 |0 |0 |0 |
8 |0 |21 |0 |0 |0 |
81 |0 |22 |0 |0 |0 |
65535 |0 |0 |0 |0 |0 |
```

3. Enter the **show restconf event-map** command to display RESTCONF event-mapping information.

```
device> show restconf event-map
SCP Event          YANG ID          YANG Mapping
153                1                /system/aaa/authentication/users
154                2                /system/aaa/server-groups/server-group/tacacs-group/
servers/server/tacacs
155                3                /system/aaa/authentication/config/authentication-method
131                4                /system/dns
143                5                /lldp/config
144                6                /lldp/interfaces
161                9                /interfaces/interface/#wc/aggregation
140                8                /interfaces/interface/#wc/ethernet/config/aggregate-id
141                7                /network-instances/network-instance/#wc/vlans
145                12               /network-instances/network-instance/#wc/protocols/protocol/
OSPF/#wc
146                13               /network-instances/network-instance/#wc/protocols/protocol/
OSPF/#wc/ospfv2/areas
147                14               /network-instances/network-instance/#wc/protocols/protocol/
OSPF/#wc/ospfv2/areas/
area/#wc/interfaces
148                15               /network-instances/network-instance/#wc/protocols/protocol/
OSPF/#wc/ospfv2/areas/
area/#wc/interfaces
149                16               /network-instances/network-instance/#wc/protocols/protocol/
STATIC/#wc/staticroutes
156                17               /system/aaa/server-groups/server-group/radius-group/
servers/server/radius
157                18               /system/aaa/authentication/icx-openconfig-aaa-aug:dot1x
158                19               /system/aaa/authorization/icx-openconfig-aaa-aug:coa/
openconfig
interfaces:interfaces
150                20               /openconfig-acl:acl
8                21               (null)
81                22               (null)
```

RESTCONF for RUCKUS ICX Switches

RESTCONF Configuration

4. Enter the **show restconf running-config** command to display RESTCONF running configuration information.

```
device> show restconf running-config

Get RC Req : File /var/log/restconf/ui_config.xml, XPath /, Def 0 State 0

RESTConf Running Configuration XML File : /var/log/restconf/ui_config.xml
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="http://yumaworks.com/ns/yumaworks-db-api">
  <interfaces xmlns="http://openconfig.net/yang/interfaces">
    <interface>
      <config>
        <description/>
        <enabled>true</enabled>
        <name>ethernet 1/1/1</name>
        <type>ianaift:ethernetCsmacd</type>
      </config>
      <ethernet xmlns="http://openconfig.net/yang/interfaces/ethernet">
        <config>
          <auto-negotiate>false</auto-negotiate>
          <duplex-mode>FULL</duplex-mode>
          <ethernet-clock xmlns="http://commscope.com/ns/yang/icx/icx-openconfig-if-ethernet
aug">none</ethernet-clock>
          <port-speed>oc-eth:SPEED_1GB</port-speed>
        </config>
      </ethernet>
      <name>ethernet 1/1/1</name>
    </interface>
    ...
  </interfaces>
  <lldp xmlns="http://openconfig.net/yang/lldp">
    <config>
      <enabled>true</enabled>
    </config>
    <interfaces>
      <interface>
        <config>
          <enabled>true</enabled>
          <name>ethernet 1/1/1</name>
        </config>
        <name>ethernet 1/1/1</name>
      </interface>
    </interfaces>
  </lldp>
</config>
```

5. Enter the **show restconf status** command to display the RESTCONF status information.

```
device> show restconf event-map
SCP Event          YANG ID          YANG Mapping
153                1                /system/aaa/authentication/users
154                2                /system/aaa/server-groups/server-group/tacacs-group/
servers/server/tacacs
155                3                /system/aaa/authentication/config/authentication-method
131                4                /system/dns
143                5                /lldp/config
144                6                /lldp/interfaces
161                9                /interfaces/interface/#wc/aggregation
140                8                /interfaces/interface/#wc/ethernet/config/aggregate-id
141                7                /network-instances/network-instance/#wc/vlans
145                11               /openconfig-spanning-tree:stp
146                12               /network-instances/network-instance/#wc/protocols/protocol/
OSPF/#wc
147                13               /network-instances/network-instance/#wc/protocols/protocol/
OSPF/#wc/ospfv2/areas
148                14               /network-instances/network-instance/#wc/protocols/protocol/
OSPF/#wc/ospfv2/areas/
area/#wc/interfaces
149                15               /network-instances/network-instance/#wc/protocols/protocol/
OSPF/#wc/ospfv2/areas/
area/#wc/interfaces
150                16               /network-instances/network-instance/#wc/protocols/protocol/
```



```
STATIC/#wc/staticroutes
156                               17           /system/aaa/server-groups/server-group/radius-group/
servers/server/radius
157                               18           /system/aaa/authentication/icx-openconfig-aaa-aug:dot1x
158                               19           /system/aaa/authorization/icx-openconfig-aaa-aug:coa/
openconfig
interfaces:interfaces
150                               20           /openconfig-acl:acl
8                                  21           (null)
81                                  22           (null)
```

High Availability

The RESTCONF protocol stack runs on active devices. On switchover, the RESTCONF protocol stack comes up on a new active node and syncs The FastIron configuration to the RESTCONF stack configuration database.

Upon failover, switchover, or reload of the active unit of the stack member, the standby unit will begin to process the RESTCONF client request.

Security Considerations

When enabled, RESTCONF services listen on port 443 and utilize TLS for security. TLS requires a CA signed certificate for both the ICX and RESTCONF clients for mutual authentication.

For ease of use, the enforcement of TLS certificates can be relaxed by the way of configuration to allow clients to present self-signed certificates.

For ICX devices that do not have a RUCKUS-signed manufacturing certificate, the RESTCONF server will present a self-signed certificate needed by TLS. Otherwise, the manufacturing device certificate will be present by default.



© 2022 CommScope, Inc. All rights reserved.
350 West Java Dr., Sunnyvale, CA 94089 USA
<https://www.commscope.com>