

RUCKUS FastIron RESTCONF Programmers Guide, 09.0.00

Supporting FastIron Software Release 09.0.00

Copyright, Trademark and Proprietary Rights Information

© 2021 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
RESTCONF for RUCKUS ICX Switches	9
Introduction.....	9
Hardware and Technology Support Matrix.....	11
RESTCONF-Supported HTTP Methods.....	11
RESTCONF Supported Modules.....	12
Logging In.....	12
URI.....	12
Base URI.....	13
Transport Protocol Requirements.....	13
Data.....	13
YANG Library Version	14
Operation Resource.....	14
CRUD Operations.....	14
POST Method	14
PUT Method.....	14
PATCH Method.....	15
DELETE Method.....	15
XML Representation.....	15
JSON Representation.....	16
Media types.....	16
Capabilities.....	16
Schema Resources.....	17
YANG Module Retrieval.....	17
Query Parameter.....	17
Web Requests with Tools like CURL.....	18
Enabling the RESTCONF Management Interface.....	18
Configuring ICX Using the RESTCONF Management Interface.....	19
Standards Conformance and RFC Support.....	21
High Availability.....	21
Security Considerations.....	21

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.ruckuswireless.com> 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://forums.ruckuswireless.com/ruckuswireless/categories>
- 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.ruckuswireless.com>.

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://training.ruckuswireless.com>.

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.

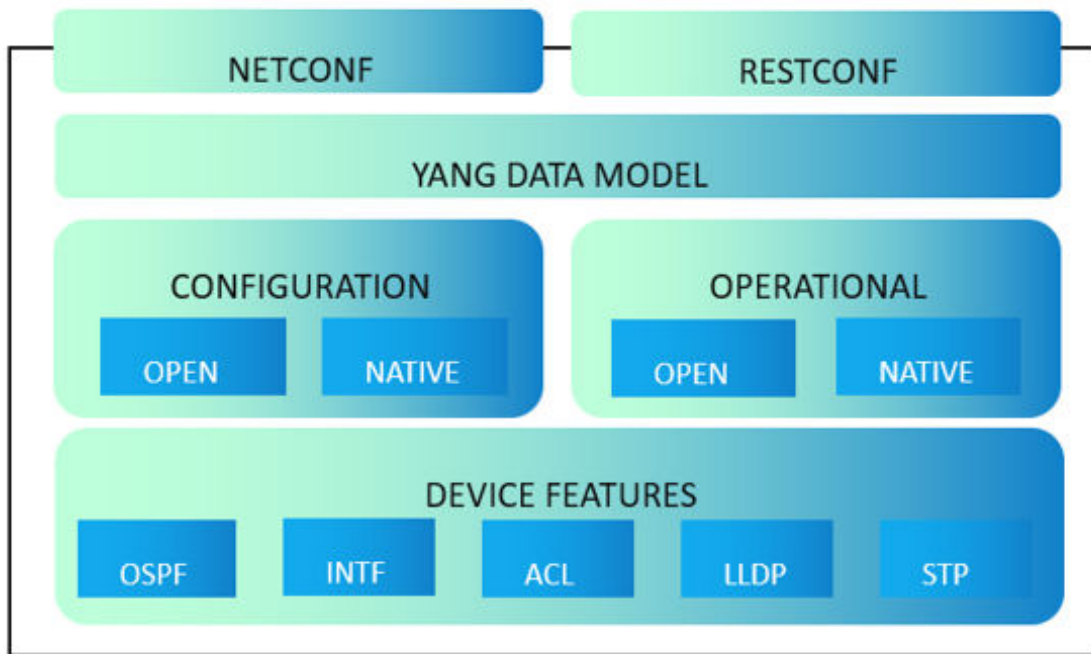
RESTCONF for RUCKUS ICX Switches

- Introduction..... 9
- Hardware and Technology Support Matrix..... 11
- RESTCONF-Supported HTTP Methods..... 11
- RESTCONF Supported Modules..... 12
- Logging In..... 12
- URI..... 12
- Base URI..... 13
- Transport Protocol Requirements..... 13
- Data..... 13
- YANG Library Version 14
- Operation Resource..... 14
- CRUD Operations..... 14
- XML Representation..... 15
- JSON Representation..... 16
- Media types..... 16
- Capabilities..... 16
- Schema Resources..... 17
- YANG Module Retrieval..... 17
- Query Parameter..... 17
- Web Requests with Tools like CURL..... 18
- Enabling the RESTCONF Management Interface..... 18
- Configuring ICX Using the RESTCONF Management Interface..... 19
- Standards Conformance and RFC Support..... 21
- High Availability..... 21
- Security Considerations..... 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 an IETF standard (RFC 6241) developed to manage network devices to retrieve and manipulate configuration data. A CLI (Command Line interface) is not the best option for automation. For software to be robust, strict data structures are required, and CLI commands and outputs are usually just strings of data without much structure. And CLI commands and outputs are usually just strings of data without much structure. NETCONF solves this problem.

YANG: Yet Another Next Generation (YANG)

REST: Representational state transfer (REST) is

The World Wide Web uses REST APIs for client/server applications. Though NETCONF is good for managing networking devices, if companies want to automate web services and networking devices, they cannot use this common tool and operators must learn two protocols for maintenance. RESTCONF was created to address this problem

RESTCONF: 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. 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.

Hardware and Technology Support Matrix

The following table lists the ICX platforms and features that support RESTCONF.

TABLE 2 RESTCONF Supported ICX Platforms and Features

Platform	Router Image	Switch Image	Stacking	Multi-Chassis Trunking (MCT)	Process Restart
ICX 7150	Yes	Yes	Yes	N/A	Yes
ICX 7250	Yes	Yes	Yes	N/A	Yes
ICX 7450	Yes	Yes	Yes	N/A	Yes
ICX 7550	Yes	Yes	Yes	N/A	Yes
ICX 7650	Yes	Yes	Yes	N/A	Yes
ICX 7850	Yes	Yes	Yes	N/A	Yes

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. The exact CLI functionally required is described at <https://jira-wiki.ruckuswireless.com/display/Team/ICX-CLI+examples>

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
LLDP	Config
Static Route	Config
OSPF	Config
ACL	Config
AAA	Config
Username / Passwords	Config
DNS	Config
STP	Config

Logging In

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:

- 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 a data type.

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 ] } } }
```

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".

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.
- 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 XML attribute, "xmlns" 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
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

JSON Representation

The ICX supports JSON format to represent the resource. This section provides information on the JSON representation for the YANG elements

- 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. Thus, 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. Thus, 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's value for the request and in 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+json	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 should maintain an entity-tag for this container and return 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 includes optional query parameters and may include other capability URIs defined outside this document.

FIGURE 2 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

Retrieval of the YANG modules is supported, 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 can be obtained. <https://%3Cmgmt%20IP%3E/restconf/data/ietf-yang-library:modules-state/module/icx-openconfig-if-poe-aug>

```
<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 using the URL provided in the schema. For icx-openconfig-if-poe-aug.yang, use the URL provided below:

<https://%3Cmgmt%20IP%3E/restconf/yang/icx-openconfig-aaa-aug/2019-09-04>

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 RFC7895.

Query Parameter

Each RESTCONF [RFC8040] operation allows zero or more query parameters to be present in the requested URI. The specific parameters that are allowed depends on the resource type, and sometimes the specific target resource used in the request.

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	Request limited subtree depth in the reply content.
fields	GET	Request a subset of the target resource contents filter GET Boolean notification filter for event stream resources.
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	Control the retrieval of default values.

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.

Web Requests with Tools like CURL

- To send 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`

Enabling the RESTCONF Management Interface

Perform this task 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.

RESTCONF protocol stack runs on separate process which is loaded with YANG models and maintains its own configuration DB wrt YANG data model. Hence FI configuration needs to be synced to RESTCONF protocol stack. The config sync is always ON if RESTCONF feature is enabled.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enable the RESTCONF management interface.

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

3. (Optional) Use the **config-sync** command to force a configuration sync 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.

```
Router(config)#restconf enable
Router(config)#restconf platform-debug-level debug
Router(config)#restconf enable-config-sync
Router(config)#restconf protocol-debug-level info
Router(config)#restconf config-sync-timeout 5
Router(config)#show rest config
```

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

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://10.176.156.24/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 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
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 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",
```

RESTCONF for RUCKUS ICX Switches

Configuring ICX Using the RESTCONF Management Interface

```
    "config": {
      "vlan-id": "100",
      "name": "VLAN 100"
    }
  }
]
}
}
$ curl -i -k -X "PATCH" https://10.176.156.24/restconf/data/network-instances/network-instance/default-vrf/
vlangs -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 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://10.176.156.24/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 request:

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

RESTCONF Example URIs

- <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>

Standards Conformance and RFC Support

RESTCONF protocol is defined in RFC 8040.

High Availability

RESTCONF protocol stack runs on active devices. On switchover, RESTCONF protocol stack comes up on new active node and syncs all FI configuration to RESTCONF stack configuration DB.

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

Security Considerations

RESTCONF services listen on port 443 when enabled 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 via 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.

