

Ruckus IoT Controller SDK Programming Guide

Copyright, Trademark and Proprietary Rights Information

© 2020 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, Ruckus Networks, Ruckus logo, the Big Dog design, BeamFlex, ChannelFly, Edgelron, FastIron, HyperEdge, ICX, IronPoint, OPENG, SmartCell, Unleashed, Xclaim, and ZoneFlex 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	4
Document Conventions.....	4
Command Syntax Conventions.....	4
Document Feedback.....	5
Ruckus Product Documentation Resources.....	5
Online Training Resources.....	5
Contacting Ruckus Customer Services and Support.....	5
Environment where Application was Tested	6
Installing Virtualbox in Ubuntu	6
Importing OVA image in Virtualbox (only once)	7
Create virtual environment for development (only once)	7
Starting application testing for dev environment (everytime)	7
Return Values of Methods	8
get_adapter().....	8
scan().....	9
connect().....	9
discover_services().....	10
read_characteristic().....	11
write_characteristic().....	11
set_char_notif_or_indication().....	11

Preface

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	device (config) # interface ethernet 1/1/6
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.

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 Ruckus products, visit the Ruckus Training Portal at <https://training.ruckuswireless.com>.

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.

Environment where Application was Tested

- 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.

Environment where Application was Tested

Operating System	Ubuntu Desktop 18.04 LTS
CPU	Intel core i3
RAM	8GB
Python	3.6.8
Virtualbox	6.0

Installing Virtualbox in Ubuntu

1. `$ sudo add-apt-repository multiverse && sudo apt-get update`
2. `$ sudo apt install virtualbox`
3. `$ sudo virtualbox`

Importing OVA image in Virtualbox (only once)

1. Open virtualbox application and click on 'File' in the menu, then select 'Import Appliance'.
2. In the popup dialog choose the path of the OVA image and click on 'Next', and keep the default settings in the next screen and click the 'Import' button.
3. Once the importing process is completed, you can see the 'VRIOT-SDK' in the left panel of virtualbox.
4. Double click on 'VRIOT-SDK' or just select it and click on 'start' in the right panel of virtualbox to start the controller.
5. Once the controller is started you will be prompted to enter
vriot login: (default value = admin)
Password: (default value = Qbc@1234)
6. After successful login, you will have a "Ruckus IoT Controller - Main Menu" where you can select choice 1 (Ethernet Network) to see the IP address assigned to the controller
7. Check whether the application has started or not from the menu by selecting choice 3 (System Operation)
8. Configure your Gateway/AP to communicate with the controller's ssl enabled MQTT broker.

Create virtual environment for development (only once)

1. Install python3 and pip3
\$ sudo apt-get -y update
\$ sudo apt-get -y upgrade
\$ sudo apt-get -y install python3-pip python3-venv
2. Open new terminal, create virtualenv
\$ python3 -m venv py3
3. Create entry in .bashrc in ubuntu
\$ echo "alias py3=' . ~/py3/bin/activate' " >> ~/.bashrc
4. Reload bash
\$ source ~/.bashrc
5. Activate virtualenv and install rpyc package
\$ py3
\$ pip install rpyc==4.1.2

Starting application testing for dev environment (everytime)

Follow the procedure to perform application testing for Dev environment

1. Get the sdkservice_wrapper and sample application in the same directory.

Return Values of Methods

2. Open new terminal and activate virtualenv

```
$ py3
```

3. Run the command `cd` to change the path to our sample application directory.
4. Edit the `sample_app.py` and change the value of 'server' variable to store controller_IP_Address.
5. Run the following command to test the sample application

```
$ python3 sample_app.py
```

If you are creating your own application follow the below instructions.

6. Import the SDKService class from `sdkservice_wrapper` into your application.
7. Create an instance of SDKService class with two arguments (server=controller_IP_Address, port=18812)

```
sdk = SDKService('controller_IP_Address', '18812')
```

8. Using this 'sdk' instance you can invoke the following methods with the arguments mentioned.

=> Arguments inside [] are optional

S.No	Method Signature	Description
1	<code>sdk.get_adapter ([gateway_id])</code>	Use this method to get available radios for the given gateway_mac. If gateway_id is not provided you will get available radios of all the gateways configured with controller IP.
2	<code>sdk.scan (gateway_id, network_mac, scan_callback, gap_name[, uuid, scan_interval, scan_window, scan_dur])</code>	Use this method to start scan with the given gateway and network mac, and you need to define a scan_callback to get the results
3	<code>sdk.connect (gateway_id, network_mac, dev_addr, addr_type, pass_key, auto_connect, connect_callback)</code>	Use this method to connect with given device, and you need to define a connect_callback to get the connection status
4	<code>sdk.discover_services(gateway_id, network_mac, dev_addr, conn_handle, service_callback)</code>	Use this method to discover the services of given device, and you need to define a service_callback to get the service results containing service and characteristic details.
5	<code>sdk.read_characteristic(gateway_id, network_mac, dev_addr, conn_handle, char_uuid, char_handle)</code>	Use this method to get the value of characteristic specified.
6	<code>sdk.write_characteristic(gateway_id, network_mac, dev_addr, conn_handle, char_uuid, char_handle, value, value_len, ack_flag)</code>	Use this method to set/write the value of characteristic specified.
7	<code>sdk.set_char_notif_or_indication(gateway_id, network_mac, dev_addr, conn_handle, char_uuid, char_handle, notif_flag, notif_callback)</code>	Use this method to set/reset notification/indication of characteristic specified if it supports notification/indication.
8	<code>sdk.disconnect(gateway_id, network_mac, dev_addr, conn_handle)</code>	Use this method to disconnect the given device with gateway.

Return Values of Methods

The return values of methods are below.

get_adapter()

Return Value

On success:

```
{
  "<gateway_id>": {
    "network_macs": [
      "<network_mac>"
    ]
  }
}
```

On Failure:

```
{"message": "description of failure"}
```

scan()

Return Value: The following are returned to specified scan_callback

On scan result:

```
{
  "<dev1_addr>": {
    "gateway_id": "<gateway_id>",
    "network_mac": "<network_mac>",
    "packet_dump": "<packet_dump>",
    "packet_type": <packet_type>,
    "dev_addr": "<dev1_addr>",
    "addr_type": <addr_type>,
    "rssi": <rssi>,
    "result": <result>
  },
  "<dev2_addr>": {
    "gateway_id": "<gateway_id>",
    "network_mac": "<network_mac>",
    "packet_dump": "<packet_dump>",
    "packet_type": <packet_type>,
    "dev_addr": "<dev2_addr>",
    "addr_type": <addr_type>,
    "rssi": <rssi>,
    "result": <result>
  }
}
```

Other cases:

```
{"message": "description of failure or scan completed or scan result empty"}
```

connect()

Return Value: The following are returned to specified connect_callback

Return Values of Methods

discover_services()

On Connection:

```
{
  "gateway_id": "<gateway_id>",
  "network_mac": "<network_mac>",
  "dev_addr": "<dev_addr>,"
  "conn_handle": <conn_handle>,
  "result": <result>
}
```

On Failure:

```
{"message": "description of connection failure"}
```

On Disconnection:

```
{"message": "device is disconnected"}
```

discover_services()

Return value: The following are returned to specified service_callback

On service result:

```
{
  "<service1_uuid>": {
    "gateway_id": "<gateway_id>",
    "network_mac": "<network_mac>",
    "service_details": {
      "conn_handle": <conn_handle>,
      "dev_addr": "<dev_addr>",
      "result": <result>,
      "service_handle": <service1_handle>,
      "service_uuid": "<service1_uuid>",
      "characteristics": [
        {
          "char_handle": <char1_handle>,
          "properties": <properties>,
          "char_uuid": "<char1_uuid>"
        },
        {
          "char_handle": <char2_handle>,
          "properties": <properties>
          "char_uuid": "<char2_uuid>"
        }
      ]
    }
  },
  "<service1_uuid>": {
    "gateway_id": "<gateway_id>",
    "network_mac": "<network_mac>",
    "service_details": {
      "conn_handle": <conn_handle>,
      "dev_addr": "<dev_addr>",
      "result": <result>,
      "service_handle": <service1_handle>,
      "service_uuid": "<service1_uuid>",
      "characteristics": [
        {
          "char_handle": <char1_handle>,
          "properties": <properties>,
          "char_uuid": "<char1_uuid>"
        },
        {
          "char_handle": <char2_handle>,
          "properties": <properties>,
          "char_uuid": "<char2_uuid>"
        }
      ]
    }
  }
}
```

Other cases:{"message": "description of failure"}

read_characteristic()

Return value:

On Success:

```
{
  "gateway_id": "<gateway_id>",
  "network_mac": "<network_mac>",
  "cmd_args": {
    "conn_handle": "<conn_handle>",
    "dev_addr": "<dev_addr>",
    "result": "<result>",
    "char_handle": "<char_handle>",
    "att_opcode": "<att_opcode>",
    "value": "<value>"
  }
}
```

On Failure:

```
{"message": "description of failure"}
```

write_characteristic()

Return value:

On Success:

```
{
  "gateway_id": "<gateway_id>",
  "network_mac": "<network_mac>",
  "cmd_args": {
    "conn_handle": "<conn_handle>",
    "dev_addr": "<dev_addr>",
    "char_handle": "<char_handle>",
    "result": "<result>",
    "ack_flag": "<ack_flag>"
  }
}
```

On Failure:

```
{"message": "description of failure"}
```

set_char_notif_or_indication()

Return value: The following are returned to specified notif_callback

Return Values of Methods

set_char_notif_or_indication()

On notified data:

```
{
  "gateway_id": "<gateway_id>",
  "network_mac": "<network_mac>",
  "cmd_args": {
    "conn_handle": "<conn_handle>",
    "dev_addr": "<dev_addr>",
    "result": <result>,
    "char_handle": <char_handle>,
    "att_opcode": <att_opcode>,
    "value": "<value>"
  }
}
```

On Failure:

```
{"message": "description of connection failure"}
```



© 2020 CommScope, Inc. All rights reserved.
Ruckus Wireless, Inc., a wholly owned subsidiary of CommScope, Inc.
350 West Java Dr., Sunnyvale, CA 94089 USA
www.ruckuswireless.com