



Application Management Commands

- [rabbitmqadmin](#), on page 2
- [smartagentctl](#), on page 5

rabbitmqadmin

Use the **rabbitmqadmin** command to check the queue status for the **RabbitAdmin** service running on the Cisco Cyber Vision Center.

rabbitmqadmin [*options*] *subcommands*

Syntax Description

-C CONFIG, --config=CONFIG	Specifies the configuration file to use. Default: ~/.rabbitmqadmin.conf
-N NODE, --node=NODE	Specifies the node described in the configuration file. Default: 'default' (only if a configuration file is specified).
-H HOST, --host=HOST	Specifies the hostname of the RabbitMQ server. Default: localhost
-P PORT, --port=PORT	Specifies the port number for the RabbitMQ HTTP API. Default: 15672
--path-prefix=PATH_PREFIX	Specifies a custom URI path prefix for the RabbitMQ HTTP API. <code>api</code> and the operation path will be appended to it. Default: blank string.
-V VHOST, --vhost=VHOST	Specifies the virtual host (vhost) to connect to. Default: All vhosts for listing commands and / for declaration commands.
-u USERNAME, --username=USERNAME	Specifies the username for authentication. Default: guest
-p PASSWORD, --password=PASSWORD	Specifies the password for authentication. Default: guest
-U URI, --base-uri=URI	Specifies a base HTTP API URI. <code>/api</code> and the operation path will be appended to it. Overrides <code>--host</code> , <code>--port</code> , and <code>--path-prefix</code> . Separately provide <code>-vhost</code> .
-q, --quiet	Suppresses status messages. Default: True
-s, --ssl	Enables SSL/TLS connection. Default: False
--ssl-key-file=SSL_KEY_FILE	Specifies the path to the PEM-formatted private key file for SSL/TLS.

<code>--ssl-cert-file=SSL_CERT_FILE</code>	Specifies the path to the PEM-formatted certificate file for SSL/TLS.
<code>--ssl-ca-cert-file=SSL_CA_CERT_FILE</code>	Specifies the path to the PEM-formatted CA certificate file for SSL/TLS.
<code>--ssl-disable-hostname-verification</code>	Disables hostname verification for SSL/TLS connections.
<code>-k, --ssl-insecure</code>	Disables all SSL/TLS validation (similar to <code>curl -k</code>).
<code>-t REQUEST_TIMEOUT,</code> <code>--request-timeout=REQUEST_TIMEOUT</code>	Sets the HTTP request timeout in seconds. Default: 120
<code>-f FORMAT, --format=FORMAT</code>	Specifies the output format.
<code>S SORT, --sort=SORT</code>	Specifies the sort key for listing commands.
<code>-R, --sort-reverse</code>	Reverses the sort order.
<code>-d DEPTH, --depth=DEPTH</code>	Specifies the maximum recursion depth for listing tables. Default: 1

Command History**Release Modification**

4.0 This command was introduced.

This example displays how to check RabbitMQ queues:

```
root@center100:~# rabbitmqadmin list queues
```

name	messages
ccv.queue.activatediscovery_scanrequest.049e52db-0178-4df6-9a63-05a04cb60736	0
ccv.queue.activatediscovery_scanresult	0
ccv.queue.authorized_sensors	0
ccv.queue.authorized_sensors.burrow	0
ccv.queue.authorized_sensors.republisher	0
ccv.queue.burrow_results	0
ccv.queue.decode_errors	0
ccv.queue.device	0
ccv.queue.dpad	0
ccv.queue.events	0
ccv.queue.flow_properties_stats	0
ccv.queue.flowtables.049e52db-0178-4df6-9a63-05a04cb60736	0
ccv.queue.flowtables.049e52db-0178-4df6-9a63-05a04cb60736.burrow	0
ccv.queue.flowtables.expired	0
ccv.queue.flowtables.unknown	0
ccv.queue.jobs.049e52db-0178-4df6-9a63-05a04cb60736	0
ccv.queue.jobs_results.aspic	0
ccv.queue.jobs_results.backend	0
ccv.queue.jobs_results.extensionapid	0
ccv.queue.jobs_results.inputd	0
ccv.queue.jobs_results.worker	0
ccv.queue.snort_events	0
ccv.queue.sysinfo	0

kraken.initial_modifications	0	
kraken.modifications	0	
kraken.queue.lc.certificate_renewal	0	
+-----+-----+		

smartagentctl

Use the **smartagentctl** command to manage the license information.

smartagentctl [**COMMAND**]

Syntax Description

<code>--register</code> <code><registration-token-file-path></code>	Registers the agent using the registration token file provided.
<code>--reregister</code> <code><registration-token-file-path></code>	Reregisters the agent using the registration token file provided.
<code>--deregister</code>	Deregisters the agent.
<code>--request-entitlement <entitlement-tag></code> <code><component-count></code>	Requests an entitlement with the specified tag and component count.
<code>--set-transport-mode <mode> [<host>]</code> <code>[<port>] [<username>] [<password>]</code>	Sets the transport mode and optionally provides connection details (host, port, username, password).
<code>--get-transport-mode</code>	Retrieves the current transport mode.
<code>--get-license-summary</code>	Retrieves a summary of the license information.
<code>--enable-reservation</code>	Enables license reservation.
<code>--disable-reservation</code>	Disables license reservation.
<code>--get-reservation-request-code</code>	Retrieves the license reservation request code.
<code>--cancel-reservation-request</code>	Cancels the license reservation request.
<code>--get-reservation-confirmation-code</code>	Retrieves the license reservation confirmation code.
<code>--get-reservation-return-code</code>	Retrieves the license reservation return code.
<code>--install-reservation</code> <code><authorization-code-file-path></code>	Installs a license reservation using the authorization code file provided.
<code>--return-reservation-authorization-code</code>	Returns the license reservation authorization code.
<code>--utility-mode-is-enabled</code>	Checks if utility mode is enabled.
<code>--enable-utility-mode</code>	Enables utility mode.
<code>--disable-utility-mode</code>	Disables utility mode.
<code>--set-customer-info <info-type></code> <code><info-data></code>	Sets customer information of the specified type.
<code>--get-customer-info</code>	Retrieves customer information.
<code>--get-tech-support</code>	Retrieves technical support information.

Command History	Release	Modification
	4.0	This command was introduced.

This example displays how to disable a license:

```
root@center100:~# smartagentctl --disable-reservation
{"status":"SUCCESS","message":"Operation Success."}
```

This example displays how to enable a license service:

```
root@center100:~# smartagentctl --enable-reservation
{"status":"SUCCESS","message":"Success"}
```

This example displays how to restart the smart agent service:

```
root@center100:~# systemctl restart sbs-backend.service
```