RabbitMQ

Installing on Windows (manual)

This guide describes how RabbitMQ can be installed and configured manually on Windows. In general we recommend using the provided Windows installer when possible.

Download the Server

Description	Download	
Binary build for Windows systems (from Bintray)	rabbitmq-server-windows-3.7.3.zip	(Signature)
Binary build for Windows systems (from GitHub)	rabbitmq-server-windows-3.7.3.zip	(Signature)

Install Erlang/OTP

First you need to install a supported version of Erlang for Windows. Run the Erlang Windows installer. Erlang will appear in the Start Menu, and \erl*x.x.x*\bin\erl.exe will be in C:\Program Files (x86) or C:\Program Files, depending on your platform and whether you chose a 32bit or 64bit version of Erlang.

Set ERLANG_HOME

If you have an existing RabbitMQ installation with the broker running as a service and you installed an Erlang VM with a different architecture then you must uninstall the service before updating ERLANG_HOME.

Set ERLANG_HOME to where you actually put your Erlang installation, e.g. C:\Program Files\erl*x.x.x* (full path). The RabbitMQ batch files expect to execute %ERLANG_HOME%\bin\erl.exe.

Go to Start > Settings > Control Panel > System > Advanced > Environment Variables. Create the system environment variable ERLANG_HOME and set it to the full path of the directory which contains bin\erl.exe.

Install RabbitMQ Server

Download rabbitmq-server-windows-3.7.3.zip from the link above.

From the zip file, extract the folder named rabbitmq_server-3.7.3 into C:\Program Files\RabbitMQ(or somewhere suitable for application files).

Synchronise Erlang Cookies

The Erlang cookie is a shared secret used for authentication between RabbitMQ cluster nodesand CLI tools. The value is stored in a file commonly referred to as the Erlang cookie file.

The cookie file used by the service account and the user running rabbitmqctl.bat must be synchronised for CLI tools such as rabbitmqctl.bat to function. All nodes in a cluster must have the same cookie value (cookie file contents).

When running a manually installed Windows service, making sure the cookie file is in sync is the responsibility of the user.

On Windows, the cookie file location varies depending on Erlang version used and whether the HOMEDRIVE or HOMEPATH environment variables are both set.

With Erlang versions starting with 20.2, the cookie file locations are:

- %HOMEDRIVE%%HOMEPATH%\.erlang.cookie (usually C:\Users\%USERNAME%\.erlang.cookie for user %USERNAME%) if both the HOMEDRIVE and HOMEPATH environment variables are set
- %USERPROFILE%\.erlang.cookie (usually C:\Users\%USERNAME%\.erlang.cookie) if HOMEDRIVE and HOMEPATH are not both set
- For the RabbitMQ Windows service
  - %USERPROFILE%\.erlang.cookie (usually C:\WINDOWS\system32\config\systemprofile)

On Erlang versions prior to 20.2 (e.g. 19.3 or 20.1), the cookie file locations are:

- %HOMEDRIVE%%HOMEPATH%\.erlang.cookie (usually C:\Users\%USERNAME%\.erlang.cookie for user %USERNAME%) if both the HOMEDRIVE and HOMEPATH environment variables are set
- %USERPROFILE%\.erlang.cookie (usually C:\Users\%USERNAME%\.erlang.cookie) if HOMEDRIVE and HOMEPATH are not both set
- For the RabbitMQ Windows service
  - %WINDIR%\.erlang.cookie (usually C:\Windows\.erlang.cookie)

To ensure Erlang cookie files contain the same string, copy the .erlang.cookie file from the Windows service directory listed above to replace the user .erlang.cookie.

Locating CLI Tools and App Data

CLI tools

Within the rabbitmq_server-3.7.3\sbin directory are some scripts which run commands to control the RabbitMQ server.

The RabbitMQ server can be run as either an application or service (not both).

- rabbitmq-server.bat starts the broker as an application.
- rabbitmq-service.bat manages the service and starts the broker.
- rabbitmqctl.bat manages a running broker.

Log in as an administrator. To see the output, run these from a Command Prompt in the sbindirectory.

Note: On Windows Vista (and later) you will need to elevate privilege (e.g. right-click on the icon to select Run as Administrator).

Set up the system path so you can find the server and sbin directory easily.

- Create a system environment variable (e.g. RABBITMQ_SERVER) for "C:\Program Files\RabbitMQ\rabbitmq_server-3.7.3". Adjust this if you put rabbitmq_server-3.7.3 elsewhere, or if you upgrade versions.
- Append the literal string ";%RABBITMQ_SERVER%\sbin" to your system path (aka %PATH%). Now you can run rabbitmq commands from any (administrator) Command Prompt.

You will need to navigate to rabbitmq_server-3.7.3\sbin to run commands if your system path does not contain the RabbitMQ sbin directory.

Data Directory

By default, the RabbitMQ logs and node's data directory are stored in the current user's Application Data directory e.g. C:\Documents and Settings\%USERNAME%\Application Data or C:\Users\%USERNAME%\AppData\Roaming.

Execute echo %APPDATA% at a Command Prompt to find this directory. Alternatively, Start > Run %APPDATA% will open this folder.

Running RabbitMQ Server as an Application

The application is started by the rabbitmq-server.bat script in sbin.

Customise RabbitMQ Server Environment Variables

Environment variable defaults are set within the rabbitmq-server.bat file. You may need to customise environment variables for your installation.

Start the Broker as an Application

Run the command

```
rabbitmq-server -detached
```

Alternatively, you can double-click the rabbitmq-server.bat file in Windows Explorer.

If you start by double-clicking, a Command Prompt window opens, displays a short banner message, concluding with the message "completed with *[n]* plugins.", indicating that the RabbitMQ broker has been started successfully.

If you started without the -detached option, e.g. by double-clicking, you will need a second Command Prompt window to control the application cleanly. *Note:* Closing the original Command Prompt window will forcefully shut down a server started this way.

Stop or Manage the Broker

rabbitmq-server only starts the broker. To manage the broker use rabbitmqctl commands.

Running RabbitMQ Server as a Service

The service will run in the security context of the system account without the need for a user to be logged in on a console. This is normally more appropriate for production use. The server should not be run as a service and application simultaneously.

The service runs using the rabbitmq-service.bat script in sbin.

Customise RabbitMQ Service Environment Variables

Note: After setting environment variables, you may need to install the service again.

The rabbitmq-service.bat script recognises many of the same environment variables as rabbitmq-server.bat, as well as a few additional service environment variables.

Install the Service

Install the service by running

```
rabbitmq-service install
```

A service with the name defined by **RABBITMQ_SERVICENAME** should now appear in the Windows Services control panel (Start > Run services.msc).

Managing the Service

To manage the service (install, remove, start, stop, enable, disable), use rabbitmq-service.bat commands. You can also use the Windows Services panel (services.msc) to perform some of the same functions as the service script.

Start the Broker as a Service

To start the broker, execute

```
rabbitmq-service start
```

If the output from this command is "Service *RABBITMQ_SERVICENAME* started", then the service was started correctly.

Confirm the service named **RABBITMQ_SERVICENAME** reports a "Started" status in Services:
Start > Run services.msc.

Port Access

Firewalls and other security tools may prevent RabbitMQ from binding to a port. When that happens, RabbitMQ will fail to start. Make sure the following ports can be opened:

- 4369: epmd, a peer discovery service used by RabbitMQ nodes and CLI tools
- 5672, 5671: used by AMQP 0-9-1 and 1.0 clients without and with TLS
- 25672: used by Erlang distribution for inter-node and CLI tools communication and is allocated from a dynamic range (limited to a single port by default, computed as AMQP port + 20000). See networking guide for details.
- 15672: HTTP API clients and rabbitmqadmin (only if the management plugin is enabled)
- 61613, 61614: STOMP clients without and with TLS (only if the STOMP plugin is enabled)
- 1883, 8883: (MQTT clients without and with TLS, if the MQTT plugin is enabled
- 15674: STOMP-over-WebSockets clients (only if the Web STOMP plugin is enabled)
- 15675: MQTT-over-WebSockets clients (only if the Web MQTT plugin is enabled)