MuntsOS

Application Note #25: RabbitMQ Enterprise Message Broker Client Programs

Revision 0
29 December 2025

by Philip Munts
***dba* Munts Technologies**
http://tech.munts.com

Introduction

This application note describes some examples of and best practices for **MuntsOS Embedded Linux** (hereafter just **MuntsOS**) target programs that send messages to and/or receive messages from a RabbitMQ Enterprise Message Broker (hereafter just **RabbitMQ**), which is often used in backend systems to implement an information bus architecture.

An information bus architecture works well for IoT (Internet of Things) networks, with IoT end nodes running software that pushes messages to and/or pulls messages from the information bus.

Hereafter, the term ***broker*** means an instance of **RabbitMQ** installed, configured, and running on a server computer accessible from a **MuntsOS** target computer. Installation and configuration of **RabbitMQ** are beyond the scope of this document.

Example Client *aka* End Node Programs

The Linux Simple I/O Library contains the following Ada and C# example **RabbitMQ** client *aka* end node programs, which can be cross-compiled to run on **MuntsOS** target computers:

- test_rabbitmq_consume.adb
- test_rabbitmq_produce.adb
- test_rabbitmq_consume
- test_rabbitmq_produce

Each of these example programs obtain their runtime configuration from the following environment variables, most of which have default values:

- `RABBITMQ_SCHEME` (default `amqp`)
- `RABBITMQ_USER` (default `guest`)
- `RABBITMQ_PASS` (default `guest`)
- `RABBITMQ_SERVER` (default `localhost`)
- `RABBITMQ_PORT` (default `5672`)
- `RABBITMQ_VHOST` (default `/`)
- `RABBITMQ_EXCHANGE` (default `amq.topic`)
- `RABBITMQ_QUEUE`
- `RABBITMQ_ROUTING` (default empty string "")

The minimum set of environment variables that you will need to define in the file `/etc/environment` on your **MuntsOS** target computer is:

- `RABBITMQ_USER`
- `RABBITMQ_PASS`
- `RABBITMQ_SERVER`
- `RABBITMQ_VHOST`

Additionally, you may need to define `RABBITMQ_ROUTING` to set the **RabbitMQ** routing key (*e.g.* topic) if the client program does not generate a custom routing key on the fly.

The LoRa radio network example producer program wioe5_ham1_rabbitmq generates a custom routing key for each incoming radio message that it passes to the broker, enabling consumer programs to select messages to and/or from particular radio nodes.