

Updated Qt5 Cross-Compilation with Ansible/Ubuntu 18.04

Cross-compiling Qt and Qt-based projects for the Raspberry Pi continues to be an error prone, complicated process. It's even worse when doing it on CentOS because of the outdated tooling (GCC 4.8?!). To lower the friction, I updated my cross-compilation Ansible playbooks to work with an Ubuntu 18.04 desktop instead of CentOS.

Background - Why Ansible?

[Ansible](#) is a tool used to automate computer configuration, usually on Linux servers. It lets an administrator run a sequence of tasks and makes sure the tasks get applied to a fleet of computers. However, it is also great for configuring just a few computers that may need some complicated configuration. If I think I might have to do a task more than once, I use Ansible instead of manually typing commands into a shell.

A good example of complicated configuration is setting up a cross-compilation toolchain on your Linux desktop so that you can build Qt apps for your Raspberry Pi. I submit to you [this excellent wiki article on setting up Qt for the Raspberry Pi](#) as an example of a complicated process. This article was indispensable to me and helped me get Qt running on my Pi2. Since my original post, [someone else wrote a step-by-step tutorial](#), showing again how complex building Qt can be.

I never want to go through the manual process again, so I wrote some Ansible playbooks to help me automate most of it. The advantage is if I have to upgrade to a future version of Qt, say 5.12.4, I'll be able to do so easily and deploy the new version to my Pi exactly like I did version 5.12.3.

John, You Already Posted This

I know. But, I rewrote all the playbooks, and this post, because I decided to use Ubuntu 18.04 instead of CentOS 7 as the cross-compilation host. I usually use CentOS (or Fedora) for my Linux projects because I prefer the Redhat-based ecosystem. But the CentOS tooling is so outdated, and Ubuntu and Raspbian match up pretty well since they're both Debian-based. I've been having a much easier time with Ubuntu.

Due to issues with the initial releases of Qt 5.12 (in particular 5.12.1) I **highly recommend using Qt >=5.12.3**, which fixes some build system bugs.

Getting Started

The [instructions have all been moved to the README on the project's Bitbucket page](#). An overview of what you need:

- Ubuntu 18.04 LTS for hosting the cross-compiler tools.
- A Raspberry Pi 2 or 3 with Raspbian Buster (the code we build is targeted at RPi2's CPU and should run on a 3).
- The playbooks. Some target the host, some target the Pi. In brief, the playbooks:
 1. Set up the host with development tools like g++, cmake, and Qt Creator.
 2. Installs development (-dev) packages on the Pi.
 3. Makes the host cross-compile a Pi version of Qt.
 4. Deploy Qt to the Pi.

A [little Ansible knowledge](#) will help if you get stuck. Flowchart of general steps to get cross-compilation working.

From:
<https://woltman.com/> - **John Woltman's Blog-o-site**

Permanent link:
<https://woltman.com/blog/2019-02-26-cross-compile-qt-for-raspberrypi-on-ubuntu>

Last update: **2021/06/21 15:03**