

如何在Ubuntu Linux 上從源代碼構建Qt 6.2.2

聆聽世界的魚 Linux公社 2021-12-07 08:30

收錄於話題

#Linux 225 #Qt 6 1 #Qt 3

[點擊上方藍字](#) • [關注Linux公社](#)

Qt 6.2.0 是Qt 6 系列的第3個版本，功能與Qt 5 相當，因此有望成為第一個被新Qt 項目或從Qt 5 移植的項目廣泛採用的版本。在這篇文章中我提供**有關在Ubuntu Linux 上從源代碼構建Qt 6.2.2的簡短教程**。

Ubuntu Desktop Linux 是使用最廣泛的桌面Linux 發行版之一，在撰寫本文時，20.04.3 版是最新的長期支持(LTS) 版本。雖然您可以使用Qt 維護工具下載預編譯的Qt 6.2.0 二進製文件，但出於多種原因，您可能更希望從源代碼構建它：

1. 您可能不想使用Qt 維護工具或創建運行它所需的Qt 帳戶。
2. 您希望生成自己的二進製文件，而不是Qt 公司提供的二進製文件。
3. 您希望在構建中包含一些源代碼補丁或其他更改。
4. 您想根據需要自定義配置。
5. 您想更好地了解從源代碼構建Qt 的過程。

一些警告。首先，**這些說明僅適用於Ubuntu Desktop Linux 20.04.3 LTS**。雖然過程應該是相似的，但其他Ubuntu 版本或Linux 發行版的依賴項和命令會有所不同。其次，**我還沒有啟用所有Qt 功能**。某些功能可能需要安裝額外的第三方依賴項才能在配置中啟用。第三：**構建需要一些時間（通常是幾個小時）**，具體取決於構建機器上可用的CPU 內核的速度和數量。

準備

要從源代碼構建所有Qt，您需要至少30GB 的可用磁盤空間，最好至少有16GB 的虛擬內存(RAM)，尤其是在構建qtwebengine 時。

有許多第三方庫和命令的依賴項。此命令應安裝必要的軟件包：

```
sudo apt install bison build-essential clang flex gperf \
libatspi2.0-dev libbluetooth-dev libclang-dev libcups2-dev libdrm-dev
libegl1-mesa-dev libfontconfig1-dev libfreetype6-dev \
libgstreamer1.0-dev libhunspell-dev libnss3-dev libopengl-dev \
libpulse-dev libssl-dev libts-dev libx11-dev libx11-xcb-dev \
libxcb-glx0-dev libxcb-icccm4-dev libxcb-image0-dev \
```

```
libxcb-keysyms1-dev libxcb-randr0-dev libxcb-render-util0-dev \
libxcb-shape0-dev libxcb-shm0-dev libxcb-sync-dev libxcb-util-dev \
libxcb-xfixes0-dev libxcb-xinerama0-dev libxcb-xkb-dev libxcb1-dev \
libxcomposite-dev libxcursor-dev libxdamage-dev libxext-dev \
libxfixes-dev libxi-dev libxkbcommon-dev libxkbcommon-x11-dev \
libxkbfile-dev libxrandr-dev libxrender-dev libxshmfence-dev \
libxshmfence1 llvm ninja-build nodejs python-is-python2 python2
```



```
linuxmi@linuxmi: ~$ sudo apt install bison build-essential clang flex gperf \
> libatspi2.0-dev libbluetooth-dev libclang-dev libcups2-dev libdrm-dev \
> libegl1-mesa-dev libfontconfig1-dev libfreetype6-dev \
> libgstreamer1.0-dev libhunspell-dev libnss3-dev libopengl-dev \
> libpulse-dev libssl-dev libts-dev libx11-dev libx11-xcb-dev \
> libxcb-glx0-dev libxcb-icccm4-dev libxcb-image0-dev \
> libxcb-keysyms1-dev libxcb-randr0-dev libxcb-render-util0-dev \
> libxcb-shape0-dev libxcb-shm0-dev libxcb-sync-dev libxcb-util-dev \
> libxcb-xfixes0-dev libxcb-xinerama0-dev libxcb-xkb-dev libxcb1-dev \
> libxcomposite-dev libxcursor-dev libxdamage-dev libxext-dev \
> libxfixes-dev libxi-dev libxkbcommon-dev libxkbcommon-x11-dev \
> libxkbfile-dev libxrandr-dev libxrender-dev libxshmfence-dev \
> libxshmfence1 llvm ninja-build nodejs python-is-python2 python2
[sudo] linuxmi 的密码:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
bison 已经是最新版 (2:3.5.1+dfsg-1)。
libatspi2.0-dev 已经是最新版 (2.36.0-2)。
libatspi2.0-dev 已设置为手动安装。
libfontconfig1-dev 已经是最新版 (2.13.1-2ubuntu3)。
libfontconfig1-dev 已设置为手动安装。
libxcb-shm0-dev 已经是最新版 (1.14-2)。
libxcb-shm0-dev 已设置为手动安装。
```

Qt 6 現在使用 *cmake* 來構建 Qt，一個問題是 Qt 6.2.0 需要 *cmake* 3.19 或更高版本，但 Ubuntu 20.04 只提供 3.16 版本。我選擇從源代碼構建最新版本的 *cmake*（在撰寫本文時為 3.21.2），使用這些命令只花了幾分鐘：

```
wget https://github.com/Kitware/CMake/releases/download/v3.21.2/cmake-
tar xf cmake-3.21.2.tar.gz
cd cmake-3.21.2
./configure
make
sudo make install
cd ..
rm -rf cmake-3.21.2
```

或者你也可以使用snap安裝，也將得到最新版本。

```
sudo snap install cmake
```

這會將較新的 *cmake* 安裝在 `/usr/local/bin` 中，即使您的Ubuntu 安裝中存在較舊的版本，Qt 構建也應該在該位置獲取它。

構建

我們現在準備構建Qt。首先，讓我們獲取源代碼：

```
wget https://download.qt.io/official_releases/qt/6.2/6.2.2/single/qt-everywhere-src-6.2.2.tar.xz
```

下載後從存檔中提取它：

```
tar xf qt-everywhere-src-6.2.2.tar.xz
cd qt-everywhere-src-6.2.2
```

現在配置，指定安裝位置。我選擇安裝在 `/usr/local/Qt6`。如果您安裝了其他Qt 版本，它們通常會放在 `/usr/local/Qt` 中，因此這不會與它們衝突。

```
./configure -prefix /usr/local/Qt6
```

這將需要一些時間，但如果存在所有必需的依賴項，應該會成功。

如果需要，可以配置或啟用更多功能，例如數據庫插件。

我們現在已準備好構建，這需要一些時間：

```
cmake --build .
```

如果它成功完成，我們現在可以將它安裝在 `/usr/local/Qt6` 中：

```
sudo cmake --install .
```

我們還想為Qt Assistant 構建文檔：

```
cmake --build . --target docs
```

然後安裝文檔：

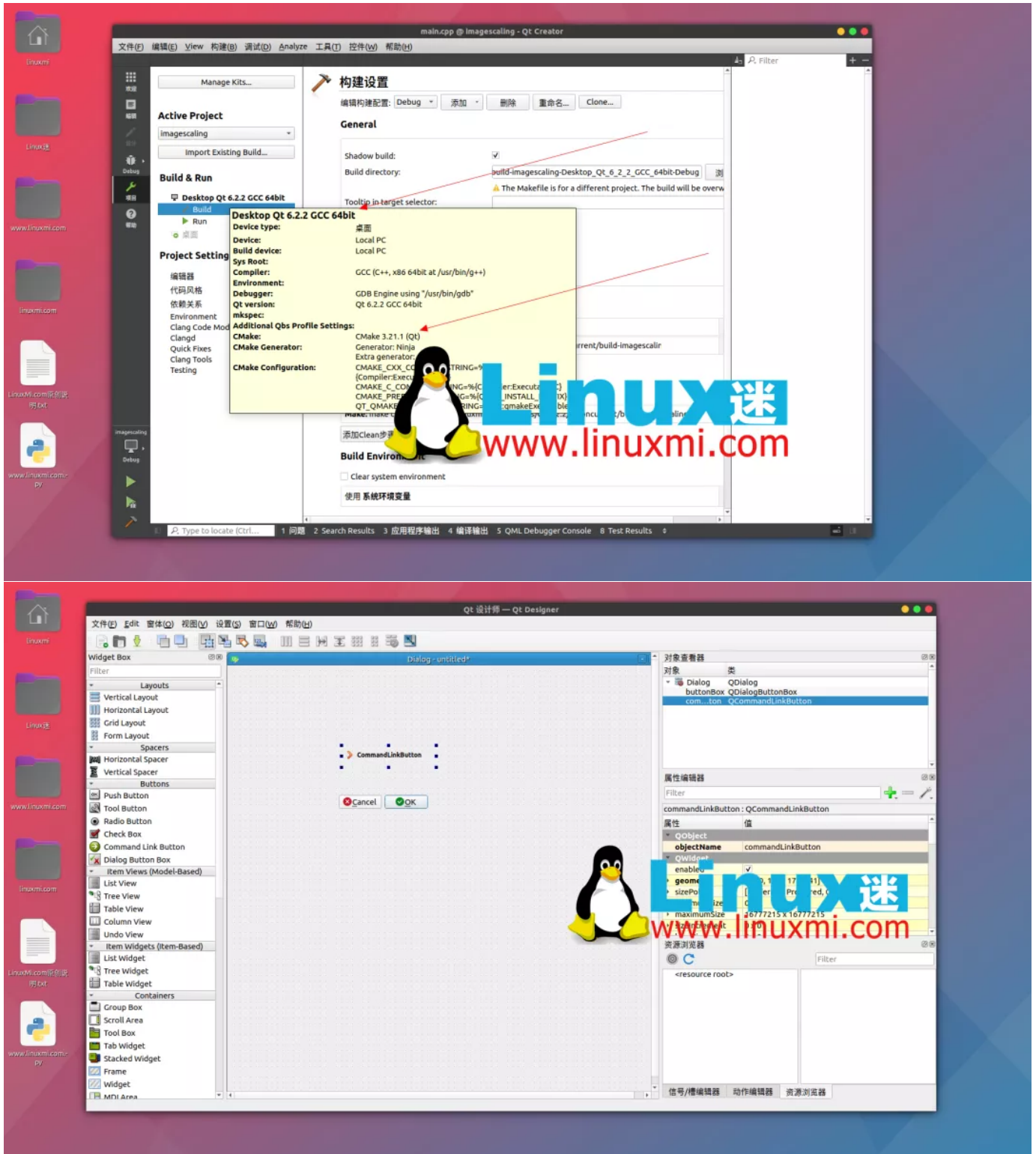
```
sudo cmake --build . --target install_docs
```

您現在可以刪除源和構建目錄。`/usr/local/Qt6` 中的實際安裝大小應該約為1 GB。

測試

安裝後，您可以對安裝進行一些基本檢查：

運行Qt Designer (/usr/local/Qt6/bin/designer) 並驗證是否出現。



將您的Qt Creator IDE 配置為指向nwq Qt 6 版本並測試您是否可以編譯和運行您自己的一些代碼。

概括

我希望你發現本教程有用。我鼓勵您嘗試從源代碼構建Qt 6，如果只是為了了解有關該過程的更多信息。如果您必須為Qt 的維護工具不直接支持的平台（例如不太常見的

Linux 發行版或嵌入式平台) 構建Qt, 則知道如何執行此操作是一項有用的技能。您可以在此處了解有關Qt 6 的更多信息。

參考

1. <https://doc.qt.io/qt-6/linux-building.html>
2. <https://doc.qt.io/qt-6/linux-requirements.html>

來自: *Linux迷*

鏈接: <https://www.linuxmi.com/qt-622-source-ubuntu-linux.html>

關注我們

長按或掃描下面的二維碼關注Linux公社



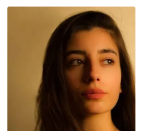
關注Linux公社, 添加“星標”
每天獲取技術乾貨, 讓我們一起成長
合作聯繫: root@linuxidc.net

閱讀原文

喜歡此內容的人還喜歡

最近和Nginx槓上了!

架構師社區



公司用的MySQL 團隊開發規範, 非常詳細, 建議收藏!

Java知音



Linux I/O 那些事兒

騰訊技術工程

