

撰写_linuxmi_于 2021年12月7日

Qt 6.2.0 是 Qt 6 系列的第3个版本，功能与 Qt 5 相当，因此有望成为第一个被新 Qt 项目或从 Qt 5 移植的项目广泛采用的版本。在这篇文章中我提供**有关在 Ubuntu Linux 上从源代码构建 Qt 6.2.2的简短教程**。

Ubuntu Desktop Linux 是使用最广泛的桌面 Linux 发行版之一，在撰写本文时，20.04.3 版是最新的长期支持 (LTS) 版本。虽然您可以使用 Qt 维护工具下载预编译的 Qt 6.2.0 二进制文件，但出于多种原因，您可能更希望从源代码构建它：

1. 您可能不想使用 Qt 维护工具或创建运行它所需的 Qt 帐户。
2. 您希望生成自己的二进制文件，而不是 Qt 公司提供的二进制文件。
3. 您希望在构建中包含一些源代码补丁或其他更改。
4. 您想根据需要自定义配置。
5. 您想更好地了解从源代码构建 Qt 的过程。

一些警告。首先，**这些说明仅适用于 Ubuntu Desktop Linux 20.04.3 LTS**。虽然过程应该是相似的，但其他 Ubuntu 版本或 Linux 发行版的依赖项和命令会有所不同。其次，**我还没有启用所有 Qt 功能**。某些功能可能需要安装额外的第三方依赖项才能在配置中启用。第三：**构建需要一些时间（通常是几个小时）**，具体取决于构建机器上可用的 CPU 内核的速度和数量。

准备

要从源代码构建所有 Qt，您需要至少 30GB 的可用磁盘空间，最好至少有 16GB 的虚拟内存 (RAM)，尤其是在构建 qtwebengine 时。

有许多第三方库和命令的依赖项。此命令应安装必要的软件包：

```

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 \
libxf86-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 \
> libxf86-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 现在使用 *cmake* 来构建 Qt，一个问题是 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。我选择从源代码构建最新版本的 *cmake*（在撰写本文时为 3.21.2），使用这些命令只花了几分钟：

```
wget https://github.com/Kitware/CMake/releases/download/v3.21.2/cmake-3.21.2.tar.gz
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安装，也将得到最新版本。

```
sudo snap install cmake
```

这会将较新的`cmake`安装在 `/usr/local/bin` 中，即使您的 Ubuntu 安装中存在较旧的版本，Qt 构建也应该在该位置获取它。

构建

我们现在准备构建 Qt。首先，让我们获取源代码：

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

下载后从存档中提取它：

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

现在配置，指定安装位置。我选择安装在 `/usr/local/Qt6`。如果您安装了其他 Qt 版本，它们通常会放在 `/usr/local/Qt` 中，因此这不会与它们冲突。

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

这将需要一些时间，但如果存在所有必需的依赖项，应该会成功。

如果需要，可以配置或启用更多功能，例如数据库插件。

我们现在已准备好构建，这需要一些时间：

```
cmake --build .
```

如果它成功完成，我们现在可以将它安装在 `/usr/local/Qt6` 中：

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

我们还想为 Qt Assistant 构建文档：

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

然后安装文档：

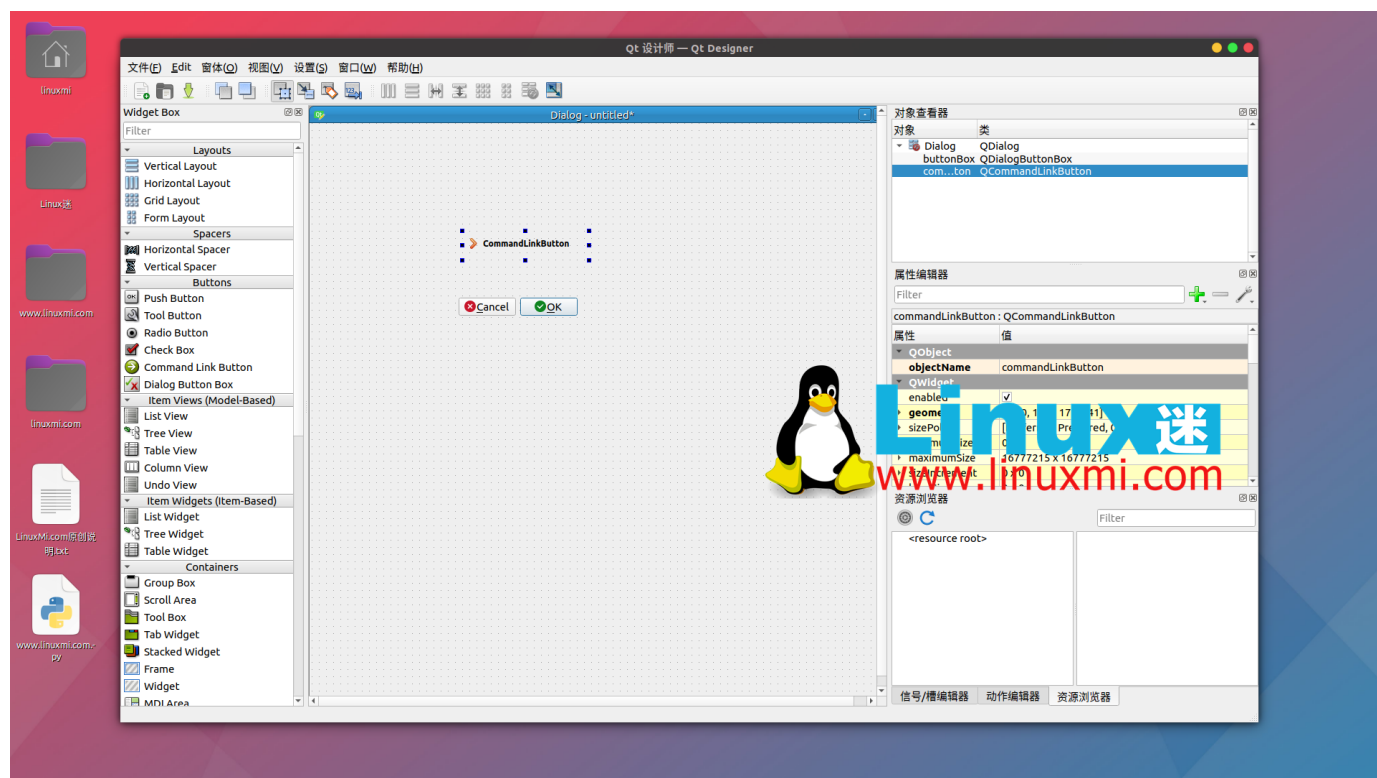
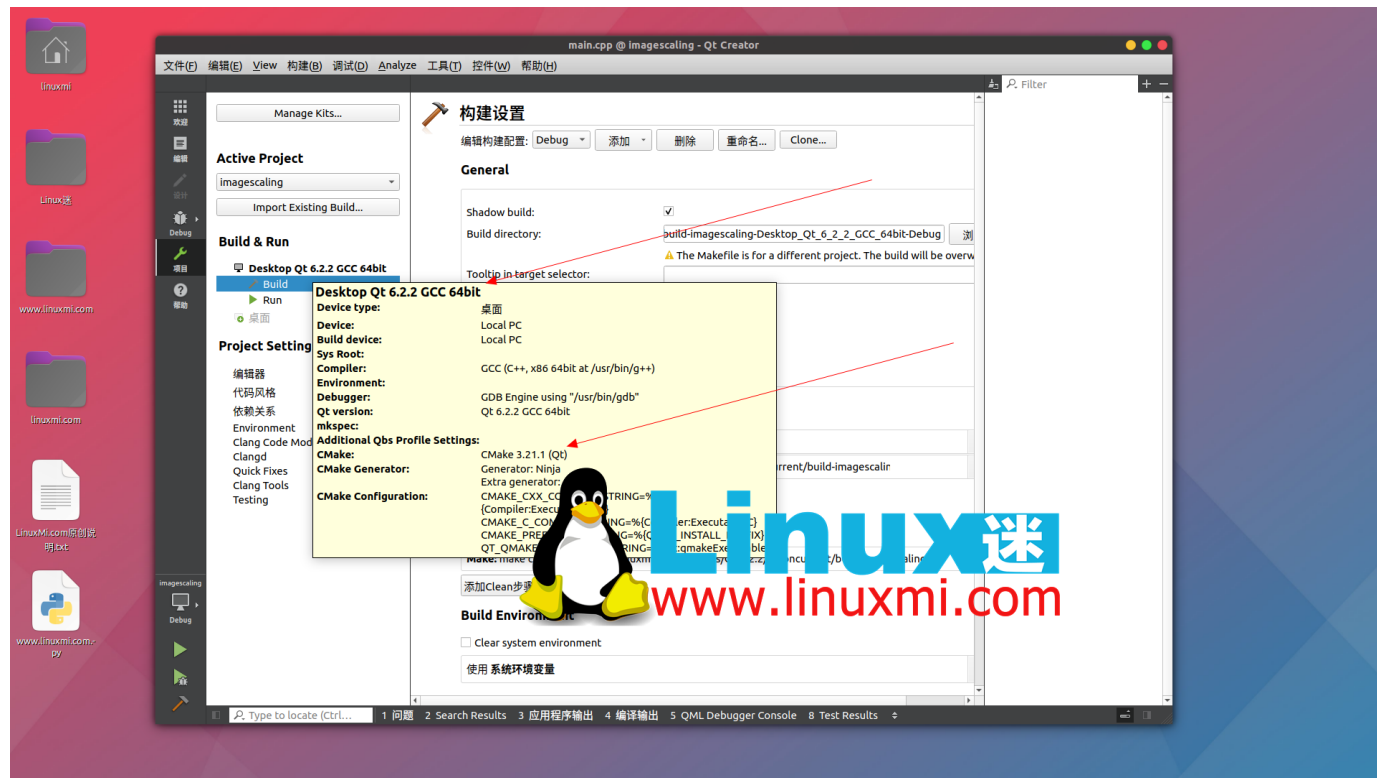
```
sudo cmake -build . -target install_docs
```

您现在可以删除源和构建目录。`/usr/local/Qt6` 中的实际安装大小应该约为 1 GB。

测试

安装后，您可以对安装进行一些基本检查：

运行 Qt Designer (/usr/local/Qt6/bin/designer) 并验证是否出现。



将您的 Qt Creator IDE 配置为指向 nwq Qt 6 版本并测试您是否可以编译和运行您自己的一些代码。

概括

我希望你发现本教程有用。**我鼓励您尝试从源代码构建 Qt 6**，如果只是为了了解有关该过程的更多信息。如果您必须为 Qt 的维护工具不直接支持的平台（例如不太常见的 Linux 发行版或嵌入式平台）构建 Qt，则知道如何执行此操作是一项有用的技能。您可以[在此处 \(https://www.qt.io/product/qt6\)](https://www.qt.io/product/qt6)了解有关 Qt 6 的更多信息。

参考

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

GoJS JavaScript Diagrams

Automatic layouts, templating, interactive

GoJS

浏览次数

[CutefishOS：媲美macOS的Manjaro Linux社区体验版](#) - 5,150 浏览

[适用于程序员和开发人员的20个最佳VS Code主题（上）](#) - 2,426 浏览

[如何在Ubuntu 20.04中启用远程桌面共享](#) - 2,247 浏览

[Ubuntu 20.04上安装PyCharm 2020专业版并激活以及设置中文](#) - 2,218 浏览

[Ubuntu 20.04 美化：安装Mac OS Catalina主题](#) - 1,983 浏览

[2021年的十大VS Code扩展，提高您的生产效率](#) - 1,803 浏览

[2020年最佳Linux桌面环境](#) - 1,710 浏览

[如何在Ubuntu 20.04上安装Gnome Tweak Tool](#) - 1,657 浏览

[如何在Ubuntu 20.04上安装/卸载XFCE](#) - 1,549 浏览

[详解 Ubuntu 20.04 安装 Qt 5.9.9 并进行环境配置](#) - 1,499 浏览

