

用qmake快速生成makefile PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E7_94_A8qmake_E5_BF_c103_144200.htm

1.简介: qmake是Trolltech公司创建的用来为不同的平台和编译器书写Makefile的工具。是qt工具包的一部分.在Unix&linux上写程式的人大概都碰过Makefile。用 make 来开发和编译程式的确很方便，可是要写出一个 Makefile就不简单了,手写Makefile是比较困难并且容易出错的，这阻挡了很多一部分的linux爱好者加入linux程序开发的阵营。虽然Open Source Software也有GNU Automake和GNU Autoconf两个软件可以生成makefile文件,但是对于一个简单的项目,使用Automake和 Autoconf就有点杀鸡也用宰牛刀了.使用qmake完全可以符合你的要求.Trolltech公司使用qmake作为Qt库和Qt所提供的工具的主要连编工具。 2.安装qmake 在linux平台上,安装完qt以及相关的qt工具包,qmake就已经被安装好了.你唯一要注意的就是QTDIR值的设定,这个必须设置到Qt被安装到的地方。如:/usr/lib/qt3/,以及qmake可执行文件的路径加到PATH路径中. 3. 一个简单的例子 用vi写个hello.c ，

```
#include <stdio.h>
int main(int argc, char** argv){printf(
" Hello, world!\n"
).return 0.}
```

 创建qmake需要的项目文件(hello.pro),

```
SOURCES = hello.cpp
CONFIG = qt warn_on release
```

 Makefile可以像这样由\ ".pro" 文件生成： `qmake -o Makefile hello.pro` 现在你的目录下已经产生了一个 Makefile 文件，输入\ " make\ " 指令就可以开始编译 hello.c 成执行文件，执行 ./hello 和 world 打声招呼吧！打开这个Makefile文件看看，是不是很专业啊！ 4.高级操作技巧 当然，在实际使用过程中，

我们的工程不可能象这个程序这样简单的，它可能有多个目录，多个头文件，多个源文件，需要链接器它不同的链接库等等情况。别急，让我和你慢慢道来。这些都是非常容易用qmake来实现的。我们从一个更加复杂的项目文件为例和你详细的讲述qmake的高级技巧：项目文件示例：
SOURCES = myqt.cpp
SOURCES = main.cpp
HEADERS = myqt.h
FORMS = xsimform.ui
TEMPLATE = lib
CONFIG = debug \\\warn_on \\\qt \\\thread \\\x11 \\\plugin
TARGET = ../bin/panel_qt
INCLUDEPATH = ../../../../xsim \\\../../../../xsim/IMdkit
DEFINES = BDB_VERSION4 \\\OS_LINUX 从这个文件可以知道

，SOURCES变量指向项目中的源文件，当项目中有多个源文件时，我们需对项目中的每一个源文件都这样做，直到结束

：SOURCES = hello.cpp
SOURCES = main.cpp当然，如果你喜欢使用像Make一样风格的语法，你也可以写成这样，一行写一个源文件，并用反斜线结尾，然后再起新的一行

：SOURCES = hello.cpp \\\main.cpp
HEADERS变量指向项目中的头文件，多个头文件的时候，和多个源文件的解决方法一致。FORMS变量指向项目中使用到的窗体文件(qtdesign设计的.ui文件)，qmake也注意了Qt的特殊需求，可以自动的包含moc和uic的连编规则。没有的话或者非qt程序可以不写。

TEMPLATE变量告诉qmake为这个应用程序生成哪种makefile。下面是可供使用的选择：
app - 建立一个应用程序的makefile。这是默认值，所以如果模板没有被指定，这个将被使用。
lib - 建立一个链接库的makefile。
vcapp - 建立一个应用程序的Visual Studio项目文件。
vclib - 建立一个库的Visual Studio项目文件。
subdirs - 这是一个特殊的模板，它可以创建一个能够

进入特定目录并且为一个项目文件生成makefile并且为它调用make的makefile。CONFIG变量指定了编译器所要使用的选项和所需要被连接的库。配置变量中可以添加任何东西，但只有下面这些选项可以被qmake识别。下面这些选项控制着使用哪些编译器标志：release - 应用程序将以release模式连编。如果\ " debug\ " 被指定，它将被忽略。debug - 应用程序将以debug模式连编。warn_on - 编译器会输出尽可能多的警告信息。如果\ " warn_off\ " 被指定，它将被忽略。warn_off - 编译器会输出尽可能少的警告信息。下面这些选项定义了所要连编的库/应用程序的类型：qt - 应用程序是一个Qt应用程序，并且Qt库将会被连接。thread - 应用程序是一个多线程应用程序。x11 - 应用程序是一个X11应用程序或库。windows - 只用于\ " app\ " 模板：应用程序是一个Windows下的窗口应用程序。console - 只用于\ " app\ " 模板：应用程序是一个Windows下的控制台应用程序。dll - 只用于\ " lib\ " 模板：库是一个共享库（dll）。staticlib - 只用于\ " lib\ " 模板：库是一个静态库。plugin - 只用于\ " lib\ " 模板：库是一个插件，这将会使dll选项生效。TARGET变量指定生成的二进制代码的路径和文件名，如果建立的是一个链接库的话，它会在文件名前面自动加上\ " lib\ " 和在最后自动加上\ " .so\ " 。我们在使用过程中可能会使用到另外的一些函数库，链接库等。函数库的头文件指定使用INCLUDEPATH变量，其它链接库的指定可以通过LIBS 变量来指定，例LIBS = -lmath -L/usr/local/libDEFINES变量的指定就如同make的-D选项一样。结束语 Autoconf 和 Automake 功能十分强大，但对于普通用户来说，太过复杂。qmake方便、简单、快捷，是一个轻量

级的makefile生成工具，虽然它是qt工具包的一部分，但它也完全可以用来进行其它程序makefile文件的生成，对于大多数人来说，它已经是非常的够用了。你也可以从qt提供的许多现存的源程序中找到相关的.pro项目文件，它们是学习qmake更多技巧的最佳范例。这篇简介只用到了 qmake 的一些皮毛罢了，希望这篇文件能帮助你产生 Makefile有个简单的依据。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com