<https://blog.csdn.net/hebbely/article/details/66970821?locationNum=1&fps=1>

简述：

在QT中，有一个工具qmake可以生成一个makefile文件，它是由.pro文件生成而来的。.pro是qmake的工程文件（project）；.pri文件可以把 \*.pro 文件内的一部分单独放到一个 \*.pri 文件内(include)，然后包含进来；.prf文件（feature），和.pri文件类似，也是被包含进.pro文件，只是更隐秘，如CONFIG+=QT；.prl文件（link），主要和生成与使用静态库密切相关。

# 1、注释

以“#”开始，到这一行结束。

快捷键：Ctrl + /

# 2、CONFIG

指定编译器选项和项目配置，值由qmake内部识别并具有特殊意义。可以指定是生成debug模式还是release模式，还是都生成。也可以用来打开编译器警告或者关闭。还可以用来配置要Qt加载库

以下配置值控制编译标志：

|  |  |
| --- | --- |
| 选项 | 说明 |
| release | 项目以release模式构建。如果也指定了debug，那么最后一个生效。 |
| debug | 项目以debug模式构建。 |
| debug\_and\_release | 项目准备以debug和release两种模式构建。 |
| debug\_and\_release\_target | 此选项默认设置。如果也指定了debug\_and\_release，最终的debug和release构建在不同的目录。 |
| build\_all | 如果指定了debug\_and\_release，默认情况下，该项目会构建为debug和release模式。 |
| autogen\_precompile\_source | 自动生成一个.cpp文件，包含在.pro中指定的预编译头文件。 |
| ordered | 使用subdirs模板时，此选项指定应该按照目录列表的顺序处理它们。 |
| precompile\_header | 可以在项目中使用预编译头文件的支持。 |
| warn\_on | 编译器应该输出尽可能多的警告。如果也指定了warn\_off，最后一个生效。 |
| warn\_off | 编译器应该输出尽可能少的警告。 |
| exceptions | 启用异常支持。默认设置。 |
| exceptions\_off | 禁用异常支持。 |
| rtti | 启用RTTI支持。默认情况下，使用编译器默认。 |
| rtti\_off | 禁用RTTI支持。默认情况下，使用编译器默认。 |
| stl | 启用STL支持。默认情况下，使用编译器默认。 |
| stl\_off | 禁用STL支持。默认情况下，使用编译器默认。 |
| thread | 启用线程支持。当CONFIG包括qt时启用，这是缺省设置。 |
| c++11 | 启用c++11支持。如果编译器不支持c++11这个选项，没有影响。默认情况下，支持是禁用的。 |
| c++14 | 启用c++14支持。如果编译器不支持c++14这个选项，没有影响。默认情况下，支持是禁用的。 |

当使用debug和release选项时(Windows下默认的)，该项目将被处理三次：一次生成一个”meta”Makefile，另外两次生成Makefile.Debug和Makefile.Release。

# 3、QT

指定项目中使用Qt的模块。默认情况下，QT包含core和gui，以确保标准的GUI应用程序无需进一步的配置就可以构建。如果想建立一个不包含Qt GUI模块的项目，可以使用“ -=”操作符。

下面一行将构建一个很小的Qt项目

QT -= gui # 仅仅使用core模块

如果要创建一个界面，里面用到XML及网络相关的类，那么需要包含如下模块

QT += core gui widgets xml network

# 4、TARGET

指定目标文件的名称。默认情况下包含的项目文件的基本名称。

TARGET = LidarPlus

上面项目会生成一个可执行文件，Windows下为LidarPlus.exe，Unix下为LidarPlus

# 5、TEMPLATE

模板变量告诉qmake为这个应用程序生成哪种makefile，可供使用的选项如下：

|  |  |
| --- | --- |
| 选项 | 说明 |
| app | 创建一个用于构建应用程序的Makefile（默认） |
| lib | 创建一个用于构建库的Makefile |
| subdirs | 创建一个用于构建目标子目录的Makefile，子目录使用SUBDIRS变量指定 |
| aux | 创建一个不建任何东西的Makefile。 |
| vcapp | 仅适用于Windows。创建一个Visual Studio应用程序项目 |
| vclib | 仅适用于Windows。创建一个Visual Studio库项目。 |

# 6、SOURCES

指定项目中的所有源文件。

SOURCES += main.cpp\

mainwindow.cpp \

serialtab.cpp \

connectorsdlg.cpp \

serialdataset.cpp \

serialotherset.cpp \

# 7、HEADERS

指定项目中的所有头文件。

HEADERS += mainwindow.h \

serialtab.h \

connectorsdlg.h \

serialdataset.h \

serialotherset.h \

# 8、FROMS

指定项目中的UI文件，这些文件在编译前被uic处理。所有的构建这些UI文件所需的依赖、头文件和源文件都会自动被添加到项目中。

FORMS += \

$$PWD/LogConsoleWidget.ui

# 9、 DEFINES

qmake添加这个变量的值作为编译器C预处理器宏（-D选项）

例如：

DEFINES += USE\_MY\_CONNECT

然后在代码中就可以使用：

#ifdef USE\_MY\_CONNECT

// TODO

#else

// TODO

#endif

# 10、RESOURCES

指定资源文件 (qrc) 的名称

RESOURCES += Resource/resource.qrc

# 11、DEPENDPATH

程序编译时依赖的相关路径.

DEPENDPATH += .

# 12、INCLUDEPATH

头文件的包含路径，即指定编译项目时应该被搜索的#include目录。如果路径包含空格，需要使用引号包含。

INCLUDEPATH = c:/include

INCLUDEPATH += "C:/extra headers"

# 13、LIBS

指定链接到项目中的库列表。如果使用Unix -l (library) 和 -L (library path) 标志，在Windows上qmake正确处理库（也就是说，将库的完整路径传递给链接器），库必须存在，qmake会寻找-l指定的库所在的目录。如果路径包含空格，需要使用引号包含路径。

win32:LIBS += c:/mylibs/math.lib

unix:LIBS += -L/usr/local/lib -lmath

# 14、DESTDIR

指定在何处放置目标文件。

CONFIG(release):DESTDIR = $$PWD/../bin

CONFIG(debug, debug|release):DESTDIR = $$PWD/../bin

# 15、MOC\_DIR

指定来自moc的所有中间文件放置的目录（含Q\_OBJECT宏的头文件转换成标准.h文件的存放目录）

# 16、OBJECTS\_DIR

指定所有中间文件.o（.obj）放置的目录。

# 17、RCC\_DIR

指定Qt资源编译器输出文件的目录（.qrc文件转换成qrc\_\*.h文件的存放目录）。

# 18、UI\_DIR

指定来自uic的所有中间文件放置的目录（.ui文件转化成ui\_\*.h文件的存放目录）。

unix:UI\_DIR = ../myproject/ui

win32:UI\_DIR = c:/myproject/ui

# 19、RC\_FILE

指定应用程序资源文件的名称。这个变量的值通常是由qmake或qmake.conf处理，很少需要进行修改。

RC\_FILE += $$PWD/UrgBenri.rc

# 20、RC\_ICONS

仅适用于Windows，指定的图标应该包含在一个生成的.rc文件里。如果RC\_FILE 和RES\_FILE变量都没有设置这才可利用。

RC\_ICONS = myapp.ico

# 21、CODECFORSRC

源文件编码方式。

CODECFORSRC = GBK

# 22、平台相关处理

根据qmake所运行的平台来使用相应的作用域来进行处理。

win32:RC\_FILE += $$PWD/UrgBenri.rc

macx:ICON = $$PWD/icons/UrgBenri.icns

win32 {

CONFIG += embed\_manifest\_exe

}

Qt4.1.3之后的版本提供了CONFIG选项来提供内嵌manifest文件的功能，embed\_manifest\_dll和embed\_manifest\_exe， 用法是将下面的选项加入pro文件， 如下：

CONFIG += embed\_manifest\_exe

默认情况下embed\_manifest\_dll已经开启。

# 23、系统变量

QMAKE\_TARGET\_COMPANY：指定项目目标的公司名称，仅适用于Windows

QMAKE\_TARGET\_PRODUCT：指定项目目标的产品名称，仅适用于Windows

QMAKE\_TARGET\_DESCRIPTION：指定项目目标的描述信息，仅适用于Windows

QMAKE\_TARGET\_COPYRIGHT：指定项目目标的版权信息，仅适用于Windows

PACKAGE\_DOMAIN：

PACKAGE\_VERSION：

RC\_CODEPAGE：指定应该被包含进一个.rc文件中的代码页，仅适用于Windows

RC\_LANG：指定应该被包含进一个.rc文件中的语言，仅适用于Windows

RC\_ICONS：指定应该被包含进一个.rc文件中的图标，仅适用于Windows

VERSION：指定程序版本号

BUILD\_NUMBER：

APP\_REVISION：

APP\_VERSION\_DATE：

APP\_VERSION：

注意：QMAKE\_TARGET\_COMPANY、QMAKE\_TARGET\_DESCRIPTION、QMAKE\_TARGET\_COPYRIGHT、QMAKE\_TARGET\_PRODUCT、RC\_CODEPAGE、RC\_LANG均仅适用于Windows， 而且只有在VERSION或RC\_ICONS 变量被设置，并且RC\_FILE和 RES\_FILE变量没有被设置的情况下生效

<http://www.cnblogs.com/lyggqm/p/6288463.html>

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/hebbely/article/details/66970821