**QT工程pro文件模板变量（TEMPLATE）**

模板变量告诉*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。

**“app”模板**

“app”模板告诉*qmake*为建立一个应用程序生成一个makefile。当使用这个模板时，下面这些*qmake*系统变量是被承认的。你应该在你的.pro文件中使用它们来为你的应用程序指定特定信息。

* HEADERS - 应用程序中的所有头文件的列表。
* SOURCES - 应用程序中的所有源文件的列表。
* FORMS - 应用程序中的所有.ui文件（由*Qt设计器*生成）的列表。
* LEXSOURCES - 应用程序中的所有lex源文件的列表。
* YACCSOURCES - 应用程序中的所有yacc源文件的列表。
* TARGET - 可执行应用程序的名称。默认值为项目文件的名称。（如果需要扩展名，会被自动加上。）
* DESTDIR - 放置可执行程序目标的目录。
* DEFINES - 应用程序所需的额外的预处理程序定义的列表。
* INCLUDEPATH - 应用程序所需的额外的包含路径的列表。
* DEPENDPATH - 应用程序所依赖的搜索路径。
* VPATH - 寻找补充文件的搜索路径。
* DEF\_FILE - 只有Windows需要：应用程序所要连接的.def文件。
* RC\_FILE - 只有Windows需要：应用程序的资源文件。
* RES\_FILE - 只有Windows需要：应用程序所要连接的资源文件。

你只需要使用那些你已经有值的系统变量，例如，如果你不需要任何额外的INCLUDEPATH，那么你就不需要指定它，*qmake*会为所需的提供默认值。例如，一个实例项目文件也许就像这样：

       TEMPLATE = app  
       DESTDIR  = c:\helloapp  
       HEADERS += hello.h  
       SOURCES += hello.cpp   
       SOURCES += main.cpp  
       DEFINES += QT\_DLL  
       CONFIG  += qt warn\_on release

如果条目是单值的，比如template或者目的目录，我们是用“=”，但如果是多值条目，我们使用“+=”来为这个类型*添加*现有的条目。使用“=”会用新值替换原有的值，例如，如果我们写了DEFINES=QT\_DLL，其它所有的定义都将被删除。

**“lib”模板**

“lib”模板告诉*qmake*为建立一个库而生成makefile。当使用这个模板时，除了“app”模板中提到系统变量，还有一个*VERSION*是被支持的。你需要在为库指定特定信息的.pro文件中使用它们。

* VERSION - 目标库的版本号，比如，2.3.1。

**“subdirs”模板**

“subdirs”模板告诉qmake生成一个makefile，它可以进入到特定子目录并为这个目录中的项目文件生成makefile并且为它调用make。

在这个模板中只有一个系统变量*SUBDIRS*可以被识别。这个变量中包含了所要处理的含有项目文件的子目录的列表。这个项目文件的名称是和子目录同名的，这样*qmake*就可以发现它。例如，如果子目里是“myapp”，那么在这个目录中的项目文件应该被叫做*myapp.pro*。

**CONFIG变量**

配置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。

例如，如果你的应用程序使用Qt库，并且你想把它连编为一个可调试的多线程的应用程序，你的项目文件应该会有下面这行：

    CONFIG += qt thread debug

注意，你必须使用“+=”，不要使用“=”，否则*qmake*就不能正确使用连编Qt的设置了，比如没法获得所编译的Qt库的类型了。

INCLUDEPATH += $${PWD}

$$表示当前pro的目录