How to Compile Qt Programs on Command line

January 13, 2006 Posted by Ravi

Qt is a cross-platform application framework (toolkit) that is widely used for developing GUI applications, and also for developing non-GUI programs such as command-line tools and consoles for servers.

Qt is the underlying library in KDE and all the KDE specific applications. Qt allows you to create robust, well designed, GUI applications in the shortest period of time.

The advantages of Qt

- Cross platform support
- Free if you are creating GPLed applications.
- Relatively less complex when compared to other libraries in the same genre.
- A unique way of communication between user interfaces using Signal - Slot mechanism.
- A rich collection of ready made widgets that reduce the development time drastically.
- A choice of Free IDEs that help you speed up the designing of your applications.

A collection of Free IDEs for Qt development is listed at the end of this article.

How to create a Qt project

Here is a cool method of creating a Qt project from a terminal.

The code shown below is a simple C++ program that will display a label with the words "Linux is wonderful" inside its own window. If you want to try it out, copy the code shown below into a text editor and save it as `test.cpp`. You can give it any name.

```
#include
#include

int main(int argc, char *argv[])
{
QApplication app(argc, argv);
QLabel *label = new QLabel("Linux is wonderful", 0);
app.setMainWidget(label);
label->show();
return app.exec();
}
```

The `.cpp` extension of the file name "test.cpp" tells us that it is a C++ file. For the uninitiated, in Linux, C++ programs are normally compiled using g++ compiler. If you don't have g++ compiler already installed, now is the right time to do so by installing the `build-essential` package which installs all the necessary compiler tools.

Usually, you use a file by name `MakeFile` which directs the compiler to compile your programs. And all you have to do is move into the directory containing the 'MakeFile' and your program, and run `make`.

So to compile the above program, you have to create a MakeFile first.

Generating a MakeFile

Qt has a easy way of generating a MakeFile. Here is how you do it.

Step 1: Move into the directory containing your code – in our case it is the file `test.cpp`.

Step 2: While in this directory, create a Qt project by running the following command –

```
$ qmake -project
```

This will create a project file called `test.pro` and include our program `test.cpp` into it.

Step 3: Now run `qmake` on the project file to create a platform specific MakeFile as follows –

```
$ qmake test.pro
```

At this stage if you do a listing of the contents of the directory, you will find a file by name MakeFile .

Compile the Qt program

To compile our program, it is now as simple as running `make`.

```
$ make
```

Executing your compiled program

After running `make`, you will find a executable file by name `test` in your directory which when run will display the label in a window.

```
$ ./test
```

This is one way of compiling a simple Qt project. For complex projects, you usually use specialised editors like those listed above which automates the creation of the project and the compilation.

Additional Resources

Qt Homepage - http://qt.digia.com/
Qt Creator - http://qt.digia.com/Product/Developer-Tools/
KDevelop - http://www.kdevelop.org/
Kommander - http://www.kde.org/applications/development/kommander/

Labels: linux , programming

7 comments:

Rune B *1/13/2006 10:20:00 PM*

For me "qmake -project" created a qt.pro file and then I had to do a "qmake qt.pro"

Nice intro to Qt. *1/17/2006 06:25:00 PM*

About the same article could be written about a young lookalike of Qt, namely Wt. But while linking against Qt gives you a native application, the result of linking against Wt gives you a web application, that renders in a web browser !

See: Wt homepage.

Make sure that the directory you create for this test is not called 'qt'. Apparently this is a known problem, as described here. *1/19/2006 09:16:00 PM*

This was very helpful... Thanks a million. Just have to compile now on windows and see what it gets me. *5/03/2008 03:17:00 PM*

shravan *8/30/2009 11:55:00 AM*

nice help for compiling a project using make file

msoos *12/01/2009 02:10:00 PM*

That is really easy. I think that you can make this as a batch file and call it from your favorite text editor (like notepad++), this article will help you doing that:

http://msoos.wordpress.com/2009/11/28/building-qt-applications-using-your-favorite-text-editor/

prabhukumara BR *1/19/2011 09:48:00 AM*

Its good ,it helps a lot to me
Thanks a lot
Here is a simple steps to write and run QT
Directory: Hi
file created inside dir: hello.cpp
run steps: qmeke -project

```
qmake Hi.pro
make
./Hi
Now it shows output.... GOOD...
```

Post a Comment

Newer Post Home Older Post

Back to Top