

Eclipse + QT 개발 환경 구축

2007년 3월 16일

CRS 3기

작업 환경

OS : Windows XP SP2

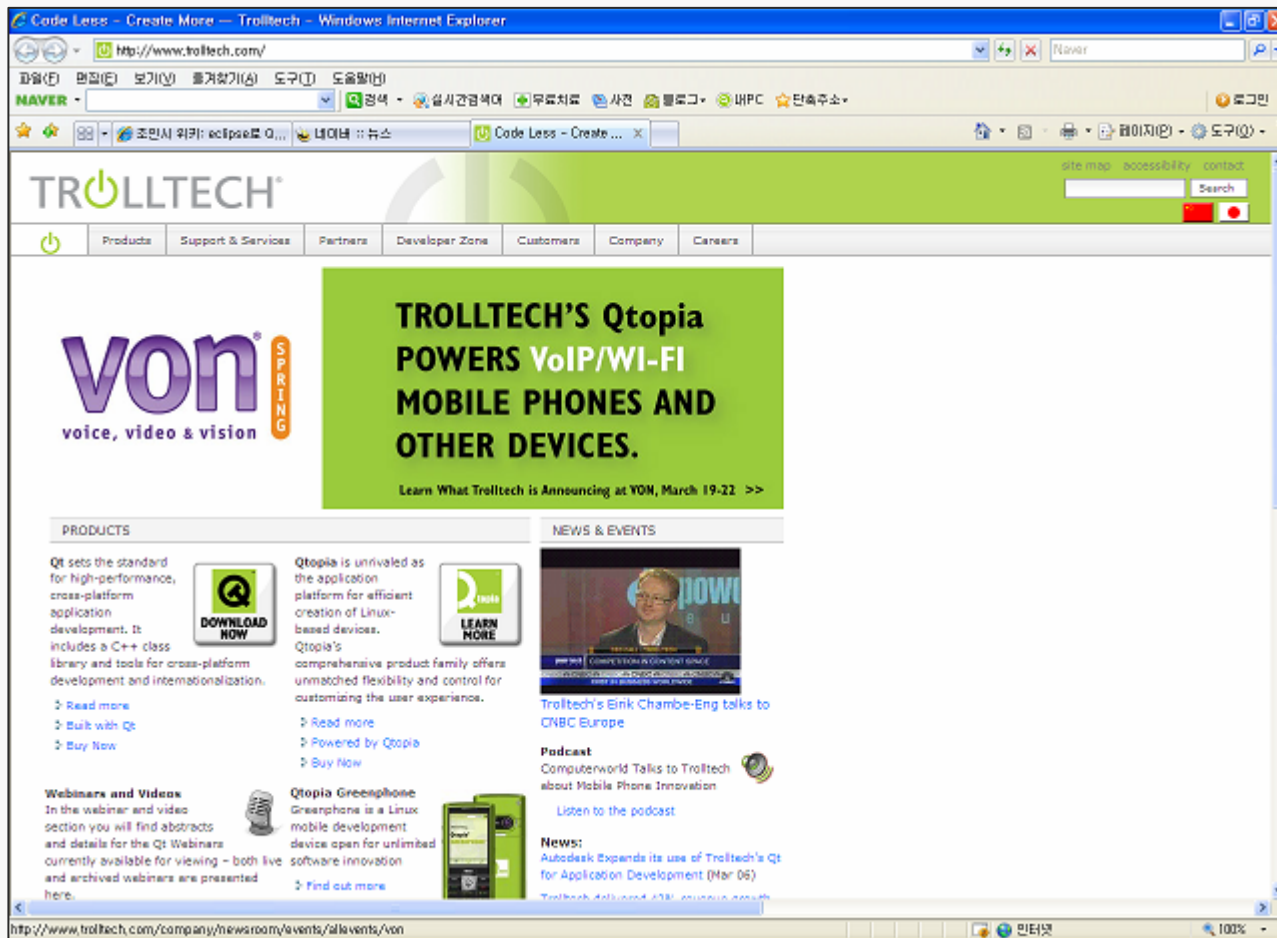
Eclipse Version : 3.2 (CDT installed)

Qt : 4.2.3 Open Source

Gcc Version : 3.4.2 (MinGW Special)

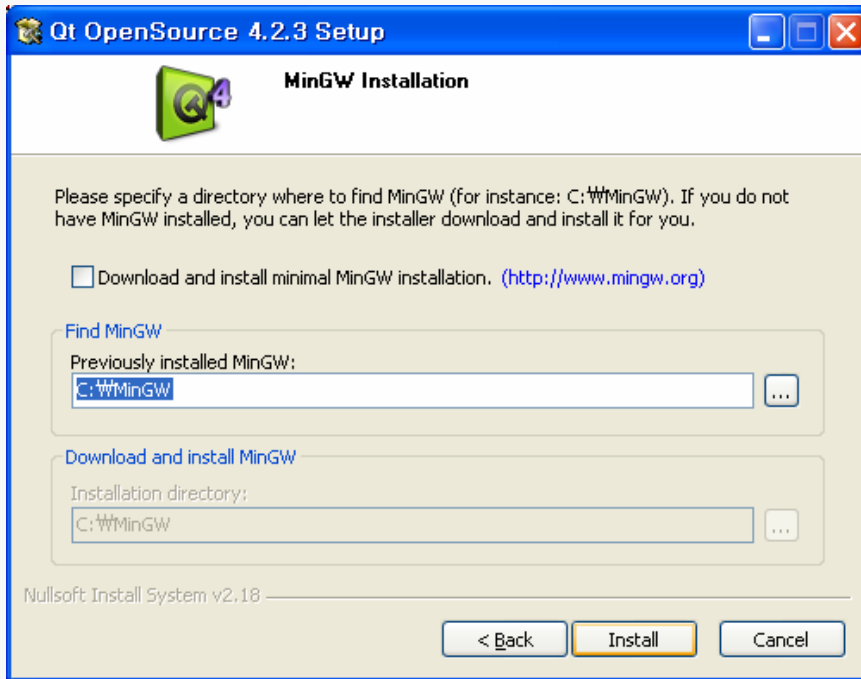
QT 4.2.3 오픈 소스 설치하기

QT Homepage : www.trolltech.com



QT Open Source for Windows

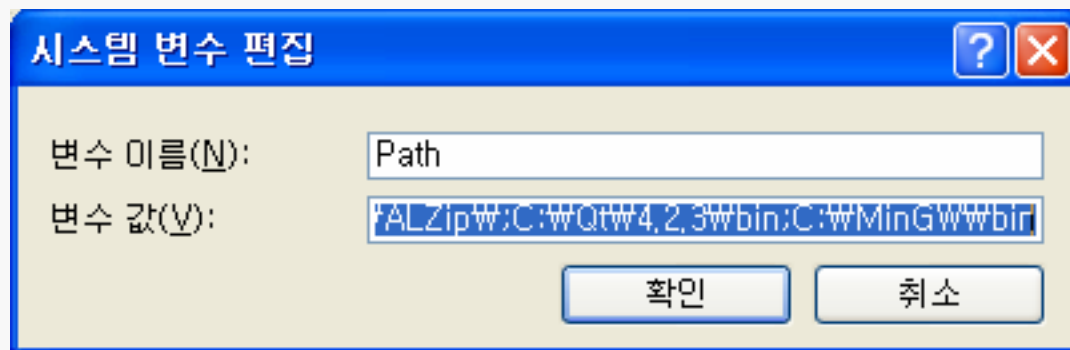
qt-win-opensource-4.2.3-mingw.exe



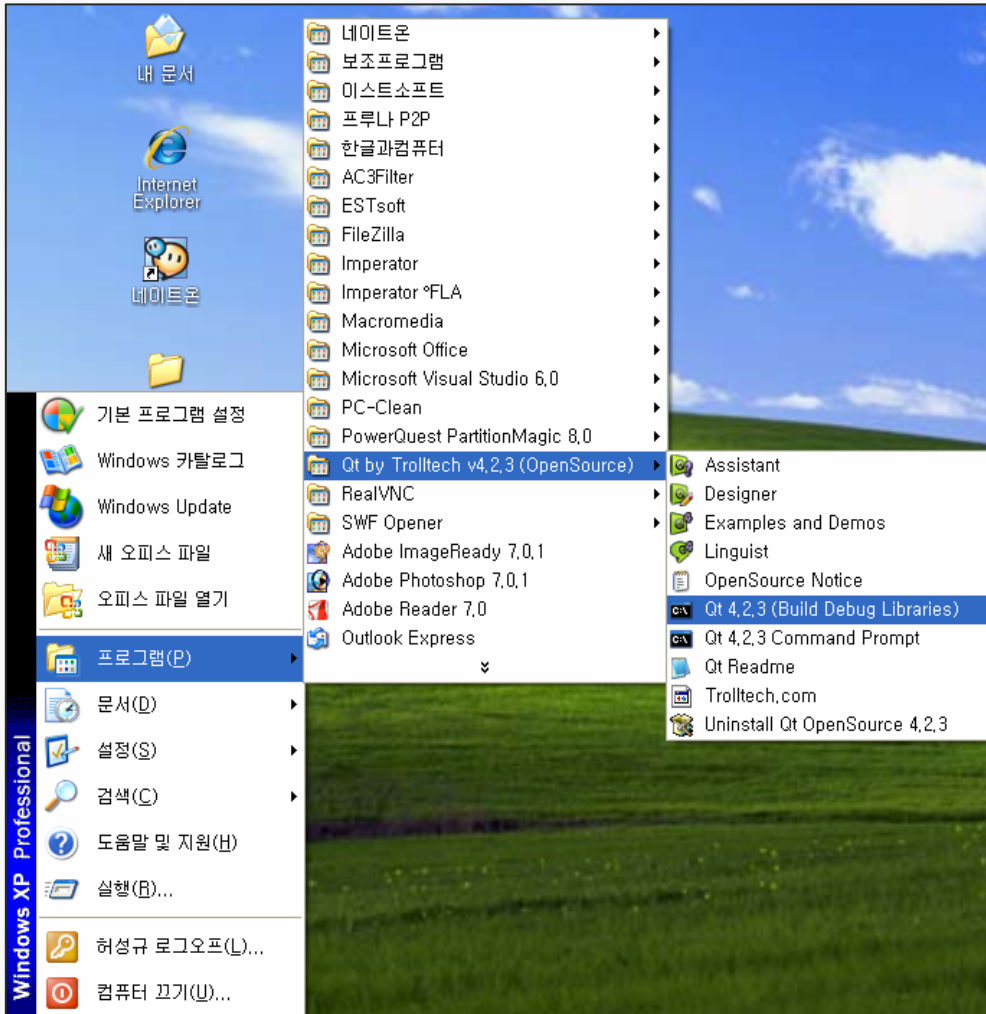
MinGW 미 설치
사용자는
Trolltech에서
제공하는
MinGW를 QT와
함께 설치 가능

QT와 MinGW 환경변수 등록

QT와 MinGW 가 설치 된 Directory내부에
있는 bin Directory를 Windows User path
또는 System path 로 등록



QT와 MinGW 연동하기



Windows의

시작 ->

프로그램 ->

Qt 메뉴의

Build Debug
Libraries

항목을 사용하여

QT환경 컴파일

(약 20~40분 소요)

Eclipse + QT 개발환경 설정

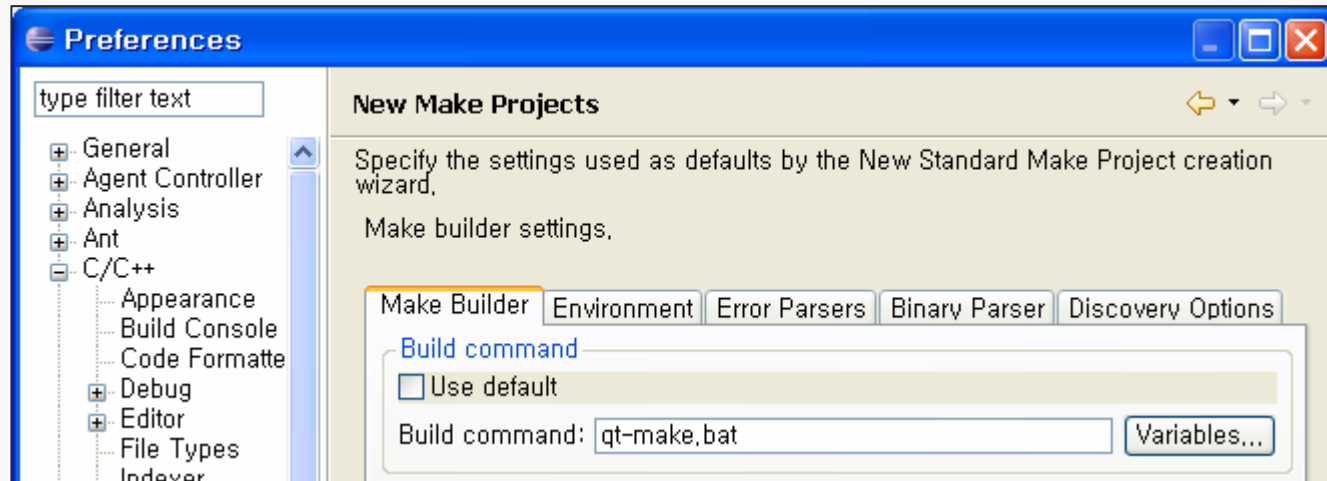
qt-make.bat

```
@echo off
IF NOT EXIST *.cc GOTO END
qmake -project
qmake
mingw32-make
:END
```

qt-make.bat 파일을 path환경변수에 등록했던
C:\Qt\4.2.3\bin Directory에 저장

main함수가 있는 소스 파일은 확장자를 .cc를 주면 됩니다. cpp 확장자를 사용하고 싶다면 위 소스의 *.cc 를 *.cpp로 수정하세요

Make 환경 설정하기(1)



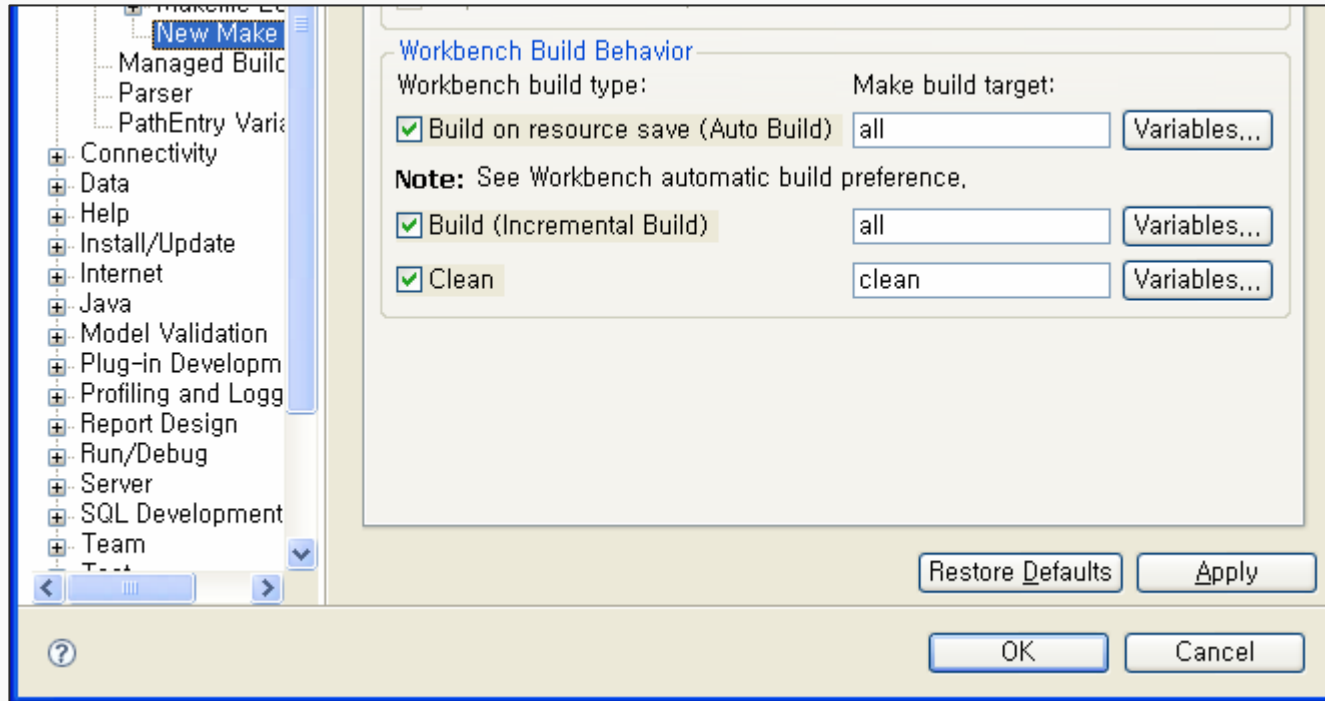
Eclipse → Window → Preferences →
C/C++ → Make → New Make Project →
Build command 항목을 qt-make.bat로 변경

위 셋팅은 Standard Make C/C++ 프로젝트에만 적용 될 뿐

Managed Make C/C++ 프로젝트와는 무관하다.

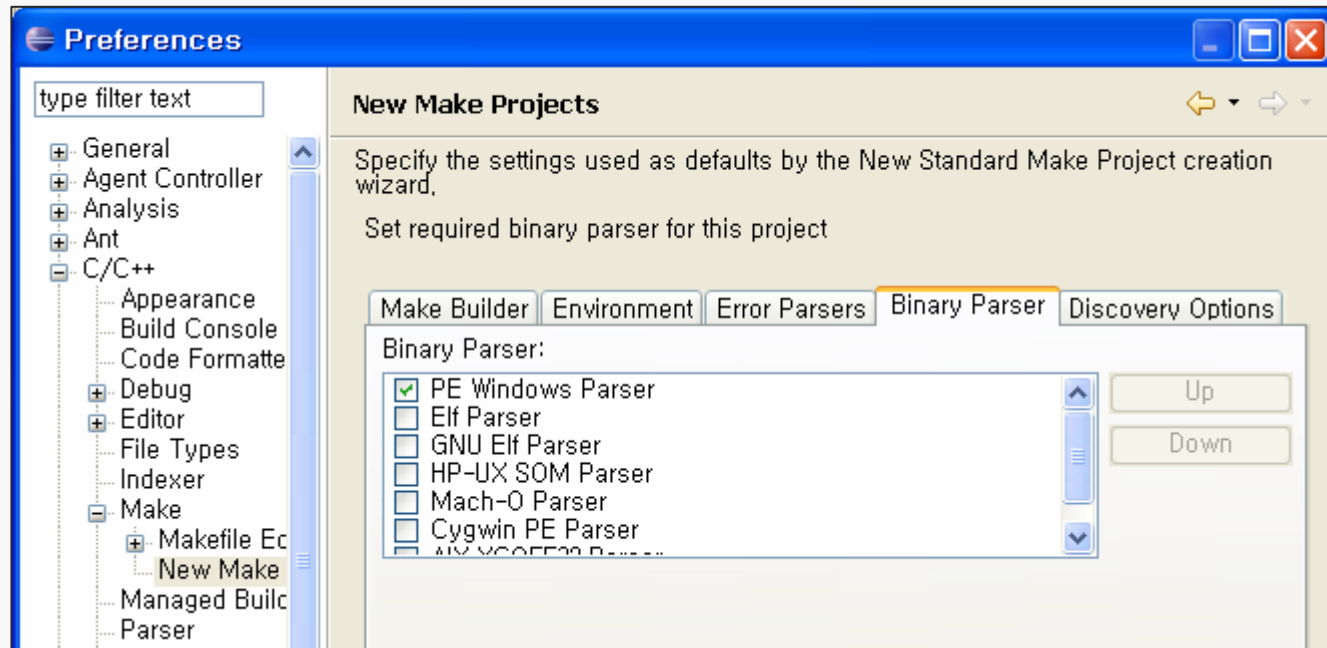
Managed 프로젝트는 Eclipse 자체적으로 setting을 관리한다.

Make 환경 설정하기(2)



Build on resource save (Auto Build) 항목을 체크해 주면 소스파일 수정 후 저장할 때 마다 자동으로 컴파일 해 주므로 편리하다.

Make 환경 설정하기(3)



Binary Parser에서 PE Windows Parser 항목을 선택하여야 한다.

예제 소스 (Standard Make C++)

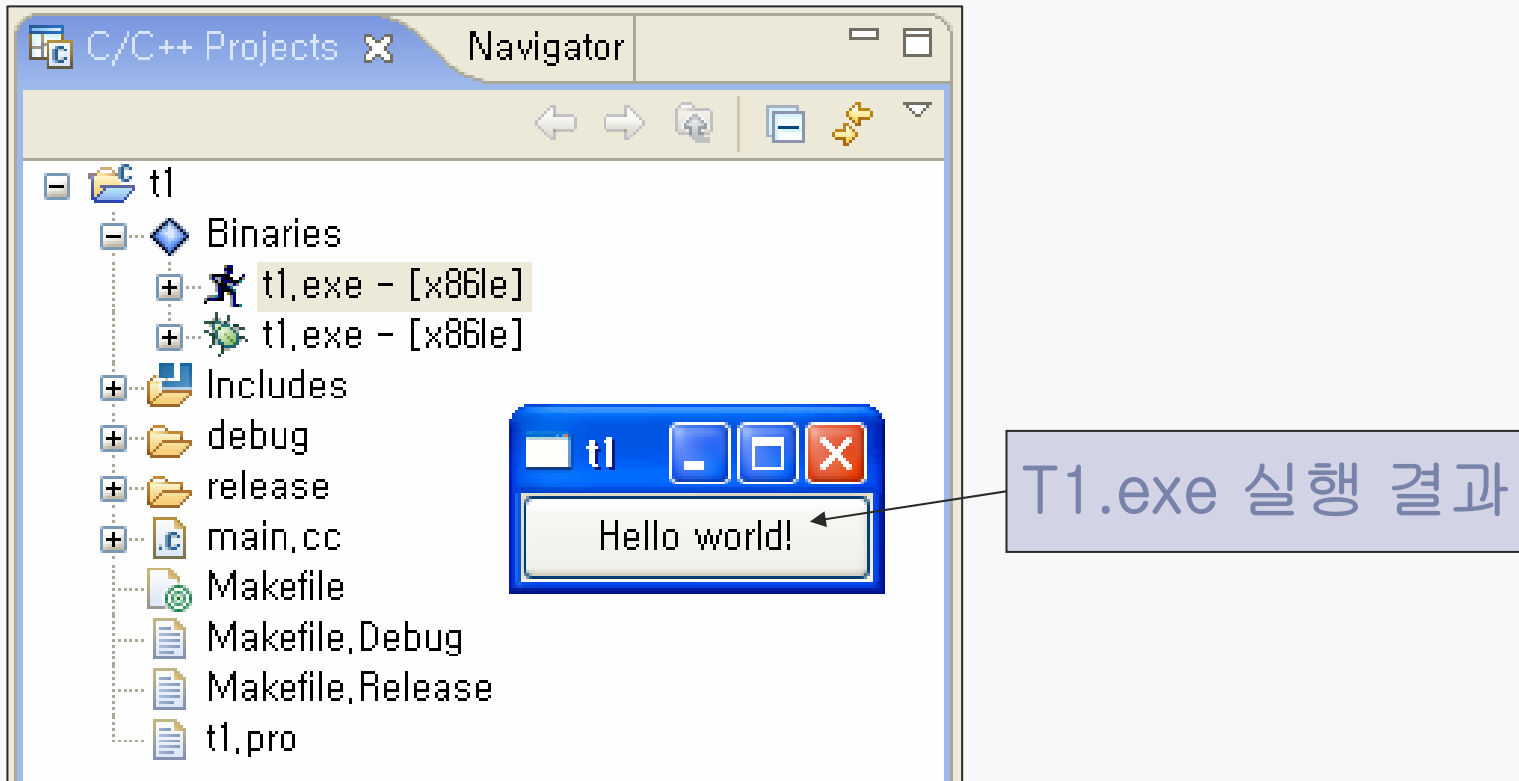
```
/* main.cc */
#include <QApplication>
#include <QPushButton>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    QPushButton hello("Hello world!");
    hello.resize(100, 30);

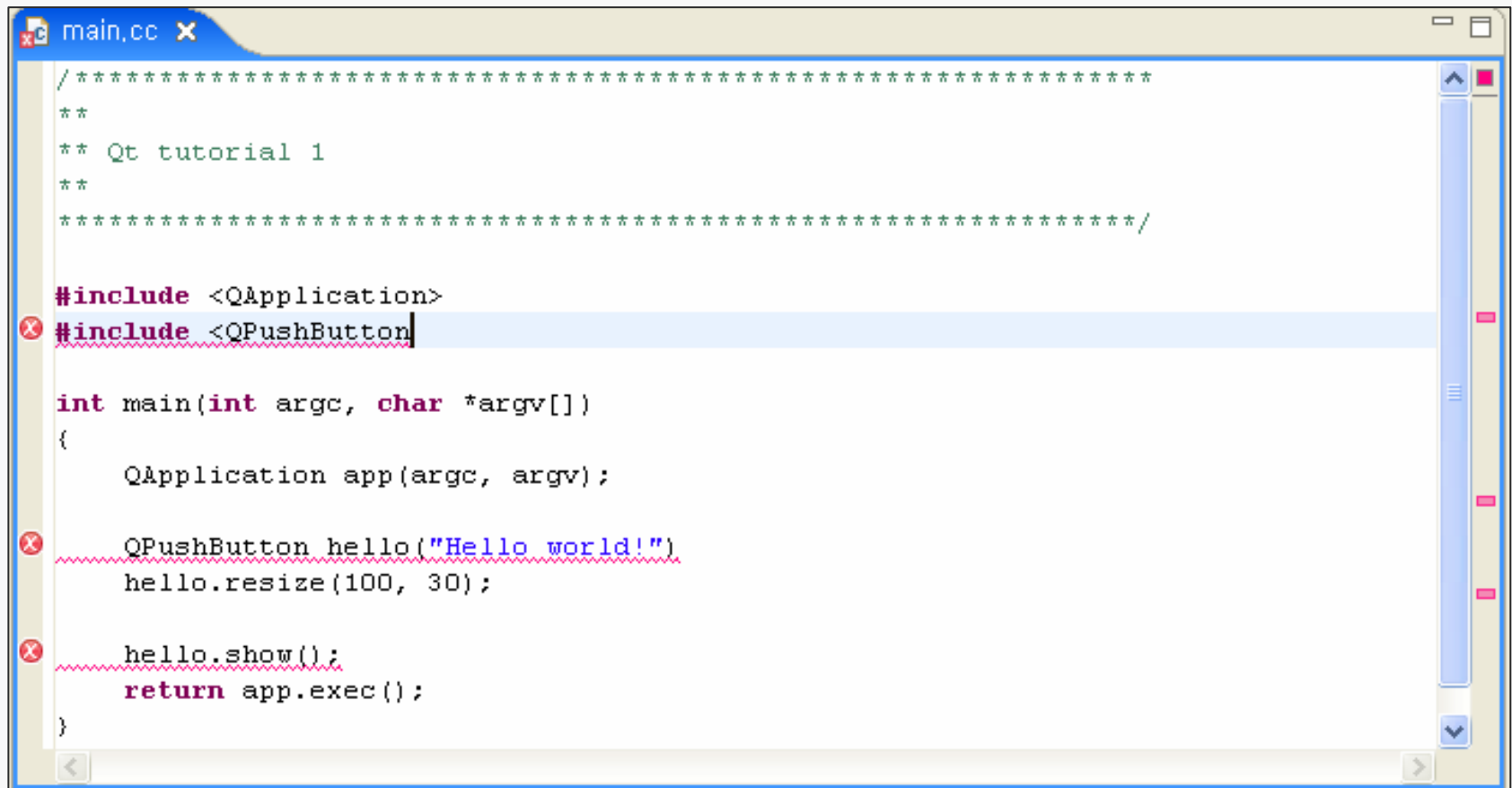
    hello.show();
    return app.exec();
}
```

예제 소스 실행 결과



실행 파일은 Debug버전과 Release버전
2개가 생성 된다.

Debugging(1)



The image shows a code editor window titled "main.cc" with a blue header bar. The code is written in C++ and includes Qt headers and a main function. There are three red 'X' error markers on the left side of the editor, indicating compilation errors. The first error is on the line `#include <QPushButton>`. The second error is on the line `QPushButton hello("Hello world!");`. The third error is on the line `hello.show();`. The code is as follows:

```
main.cc x
/*****
**
** Qt tutorial 1
**
*****/

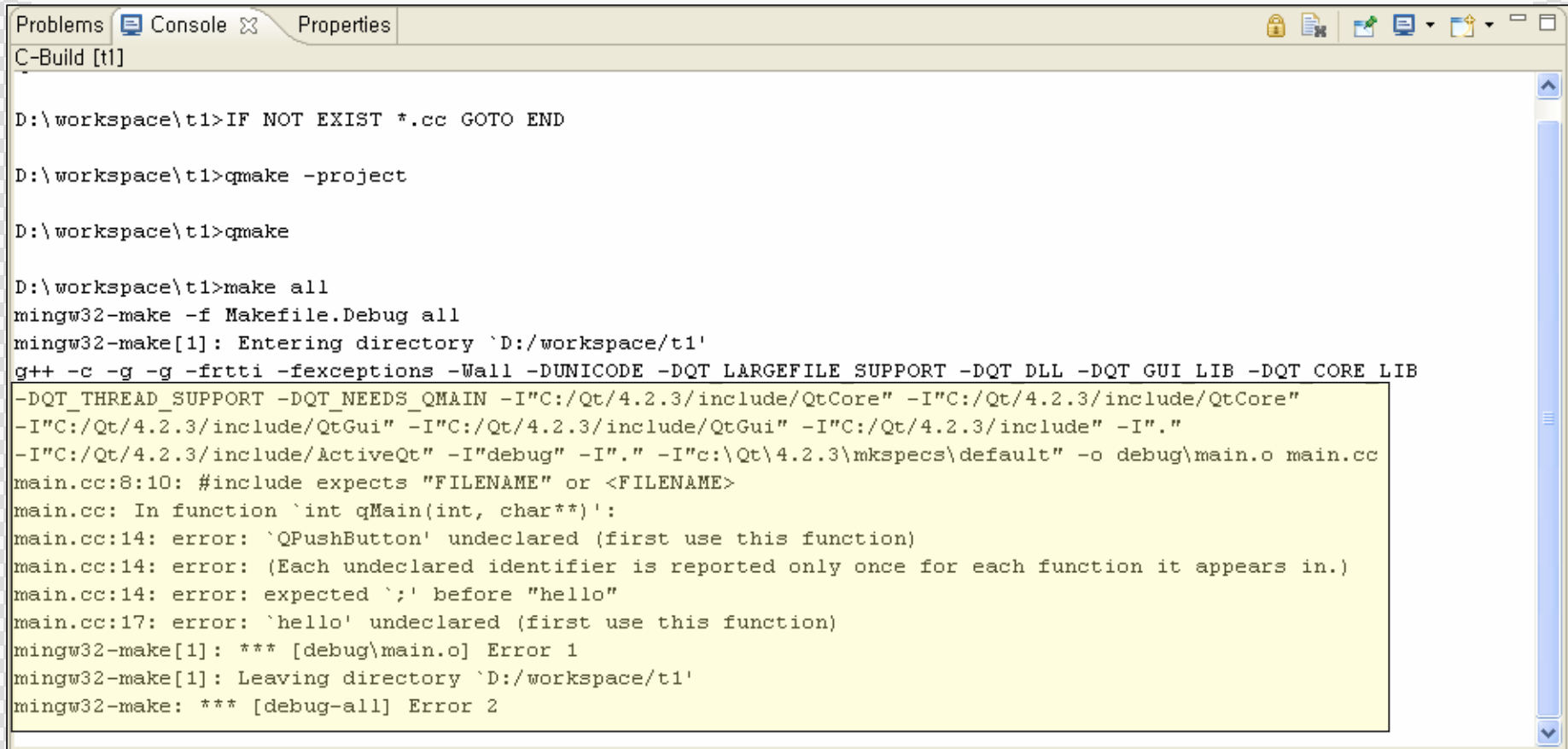
#include <QApplication>
#include <QPushButton>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    QPushButton hello("Hello world!");
    hello.resize(100, 30);

    hello.show();
    return app.exec();
}
```

Debugging(2)



```
Problems Console Properties
C-Build [t1]
D:\workspace\t1>IF NOT EXIST *.cc GOTO END
D:\workspace\t1>qmake -project
D:\workspace\t1>qmake
D:\workspace\t1>make all
mingw32-make -f Makefile.Debug all
mingw32-make[1]: Entering directory `D:/workspace/t1'
g++ -c -g -g -frtti -fexceptions -Wall -DUNICODE -DQT_LARGEFILE_SUPPORT -DQT_DLL -DQT_GUI_LIB -DQT_CORE_LIB
-DQT_THREAD_SUPPORT -DQT_NEEDS_QMAIN -I"C:/Qt/4.2.3/include/QtCore" -I"C:/Qt/4.2.3/include/QtCore"
-I"C:/Qt/4.2.3/include/QtGui" -I"C:/Qt/4.2.3/include/QtGui" -I"C:/Qt/4.2.3/include" -I"."
-I"C:/Qt/4.2.3/include/ActiveQt" -I"debug" -I"." -I"c:\Qt\4.2.3\mkspecs\default" -o debug\main.o main.cc
main.cc:8:10: #include expects "FILENAME" or <FILENAME>
main.cc: In function `int qMain(int, char**)':
main.cc:14: error: `QPushButton' undeclared (first use this function)
main.cc:14: error: (Each undeclared identifier is reported only once for each function it appears in.)
main.cc:14: error: expected `;' before "hello"
main.cc:17: error: `hello' undeclared (first use this function)
mingw32-make[1]: *** [debug\main.o] Error 1
mingw32-make[1]: Leaving directory `D:/workspace/t1'
mingw32-make: *** [debug-all] Error 2
```

그 외 사항들..

- ◆ Windows Vista 환경에서는 Debug Library Build가 수행되지 않음(한대의 컴퓨터에 Xp와 Vista를 같이 사용할 시에는 Xp에서 Build 후 Vista에서 작업 가능)
- ◆ 시작->Qt메뉴의 Assistant 와 Designer 를 이용한 프로그래밍은 MFC의 성능과 거의 대등한 것 같음
- ◆ Trolltech홈페이지에서 리눅스 환경용 소스파일을 받아 위와 같은 방법(qt-make.bat 파일 제외)으로 설치 가능하며 잘 작동함(추후 업데이트)
- ◆ Eclipse에서 workspace 변경시 preference부분 setting 을 다시 해 주어야 함
- ◆ Cygwin이 설치 되어 있다면 Path 환경변수에서 MinGW 값을 우선으로 할당하여야 함