

# MinGW, wxWidget, Code:Blocks를 이용한 C/C++ IDE 환경 구축

지은날 : 2006년 5월 16일

판올림 : 2006년 5월 25일

지은이 : 이민우

필자는 십여년간 VC++를 이용하여 윈도우 응용프로그램을 개발해 왔다. 그러나 리눅스에 관심을 갖게 되면서 오픈 소스 프로젝트들에도 관심을 갖게 되었고, 사회 공용 자원들에 대한 배타적인 소프트웨어들에 대한 대안을 찾는게 취미가 되었다.

리눅스에서는 이런 오픈 소스 프로젝트들이 많이 활성화 되었지만, 윈도우에서는 여전히 배타적 라이선스를 갖는 상용 프로그램들이 널리 사용되는 것에 대해서 안타까운 생각을 갖고 있으며, 특히 윈도우 응용 프로그램을 개발하기 위해선 Visual Studio에 전적으로 의존할 수 밖에 없는 상황을 개척해 보고자, 여러 가지 방법을 찾던 중, Eclipse + CDT + MinGW + wxWidget을 이용한 방법을 시도해 보았으나, Eclipse가 자바 기반으로 동작되다 보니 유용하고 뛰어난 기능이 많음에도 불구하고, 너무나 많은 메모리를 차지하고 시스템의 전반적인 성능을 매우 떨어뜨리는 상황을 극복할 뚜렷한 대안이 없어, 새로운 방법을 찾게 되었다. 그 유력한 대안이 바로 이 글에서 다루고자 하는 것이다.

이 방법은 윈도우 뿐만 아니라 리눅스, 매킨토시에서도 적용이 가능하다. 리눅스에서 구축 방법은 이 문서의 마지막에서 다루겠다.

## IDE 구축에 필요한 패키지들

- ◆ MinGW 컴파일러 : 윈도우용 컴파일러(gcc, make, binutils)
- ◆ wxWidget : GUI 라이브러리
- ◆ Code::Blocks : 통합 개발 환경

## 설치 전 요구 사항

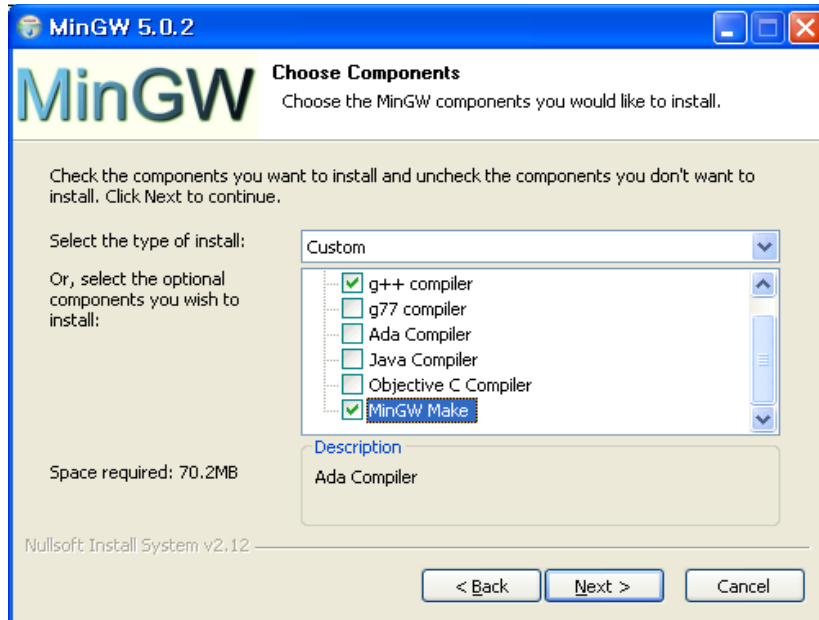
먼저 IDE를 설치할 기본 디렉토리를 만든다. 필자의 경우, C:\WIDE에 만들었다. 이 디렉토리는 문서 전체에 걸쳐 <IDE\_ROOT>로 대체되어 표시된다.

## 1. MinGW 설치

Code::Blocks 는 MinGW를 포함한 배포판과 순수 Code::Blocks 배포판을 제공한다. 전자의 경우,

MinGW를 따로 설치할 필요가 없으나, 필자는 패키지별 독립성을 훼손하지 않기 위하여 각 패키지별로 별도로 설치하였다.

- ◆ MinGW 내려받기 사이트에서 최신의 MinGW 파일을 내려받아서 <IDE\_ROOT> 밑에 설치한다.
- ◆ 내려받기 주소는 <http://prdownloads.sf.net/mingw/MinGW-5.0.2.exe?download> 이다.
  - 여러가지 파일들이 있으나, MinGW-5.0.2.exe 에 우리가 필요한 대부분의 파일들이 들어있다.
- ◆ 내려받은 MinGW-5.0.2.exe 프로그램을 저장할 임의의 폴더를 하나 만든 후, 그 곳에 이 파일을 저장한다.
- ◆ 프로그램을 실행하면, 압축되어 포함된 각종 패키지들이 압축 해제된다.
  - "MinGW-5.0.2.exe"를 실행하고, 설치 옵션에서 "Custom"을 선택하고,
  - "MinGW base tools", "g++ compiler", "MinGW Make" 구성 요소를 선택한다.
  - 설치 디렉토리에 <IDE\_ROOT>WMinGW를 선택한 다음, 나머지는 설치 과정에 따른다.



이전 버전의 MinGW에서는 gdb가 포함되어 있었으나, MinGW-5.0.2에는 gdb가 포함되어 있지 않다. 따라서 gdb를 내려받아 MinGW가 설치된 디렉토리에 설치하여야 한다.

- ◆ 내려받기 주소는 <http://prdownloads.sf.net/mingw/gdb-6.3-2.exe?download> 이다.
    - 설치 화면에서 설치 디렉토리에 <IDE\_ROOT>WMinGW 를 선택하고, 나머지는 설치 과정에 따른다.
  - ◆ 성공적으로 설치가 끝나면 PATH 환경 변수에 다음 디렉토리를 등록해 줘야 한다.
    - "제어판" -> "시스템" -> "고급" -> "환경 변수" -> "시스템 변수" 에서 PATH에 추가 한다.
- ☞ <IDE\_ROOT>WMinGW\bin;

## 2. wxWidgets 설치

◆ <http://www.wxwidgets.org> 사이트에서 wxWidgets을 내려받는다.

◆ 내려받을 프로그램은 <http://prdownloads.sourceforge.net/wxwindows/wxMSW-2.6.3-Setup-1.exe> 이다.

◆ 위 프로그램을 실행하면 설치 화면이 나오는데, 설치 디렉토리만 <IDE\_ROOT>WwxWidgets-2.6.3 로 설정해주고 나머지는 설치 과정에 따른다.

◆ wxWidgets 패키지가 성공적으로 설치되었으면, 사용자 환경 변수에 **WXWIN** 이 등록되어 있을 것이다. 이것을 시스템 변수로 옮긴다(굳이 안해도 상관없으나 필자의 경우, 전역적 환경을 구축하고자 변경하였다).

필자가 이 패키지를 설치하고, 조금 해했는데, 그 이유는 이 패키지는 소스 형태로 배포되고, 라이브러리 라이브러리들을 제공하지 않는 것을 몰랐기 때문이다. 따라서, 패키지 설치가 끝나면 "1. MinGW 설치" 에서 설치한 컴파일러를 이용하여 라이브러리들을 컴파일해야 한다.

◆ 먼저 도스창을 하나 연다..

◆ make 버전을 다음 명령으로 확인한다.

- Prompt> `mingw32-make -v` # GNU make 3.80인지 확인한다.

## 2.1 wxWidgets 빌드

◆ 이제 wxWidgets를 컴파일하기 위하여 wxWidgets 디렉토리로 이동한다.

- Prompt> `cd C:`

- Prompt> `cd %WXWIN%\WbuildWmsw`

◆ 먼저 소스에 대한 크린업을 수행한다(아래 명령들은 모두 한 줄로 이어서 입력하여야 한다).

- Prompt> `mingw32-make -f makefile.gcc MONOLITHIC=1 SHARED=1 UNICODE=1`

**BUILD=release clean**

주:) 옵션이 유니코드에 릴리즈 빌드라는 점에 주목하라.

◆ 모든 것이 깨끗해졌으면 이제 컴파일을 시작한다.

- Prompt> `mingw32-make -f makefile.gcc MONOLITHIC=1 SHARED=1 UNICODE=1`

**BUILD=release**

주:) 여기 사용된 옵션들은 Code::Blocks에서 공식적으로 지원되는 옵션이며, 다른 옵션들을 원한다면 Code::Blocks를 재구성하여야 하고, 그러기 위하여 또다른 삽질이 필요하다.

이 과정은 꽤 시간이 걸린다. 이 시점에서 커피나 한잔 마시자. 나의 팬티엄-M 1.5 GHz 노트북에서 30 분 정도 걸렸다.

### 3. Code::Blocks 설치

◆ <http://codeblocks.org> 사이트에서 "Code::Blocks IDE, without compiler" 윈도우용 바이너리를 내려 받는다.

◆ 내려받을 프로그램은 <http://prdownloads.sourceforge.net/codeblocks/codeblocks-1.0rc2.exe?download> 이다.

- 이 바이너리 배포판은 기능상 문제가 있기 때문에 실제 사용하지는 않지만, 기본 설정 상태를 유지하기 위하여 설치한다.

◆ codeblocks-1.0rc2.exe를 실행하고, 옵션에서 모든 항목을 선택하되, SVN/CVS support for projects 는 제외한다(아직 SVN/CVS 지원 기능에 버그가 있다).

◆ 설치 디렉토리를 <IDE\_ROOT>WCodeBlocks 로 하고, 나머지는 설치 과정에 따른다.

◆ 최근의 Nightly Build 패키지를 내려 받는다.

- <http://forums.codeblocks.org/index.php?board=20.0> 로부터 최근의 빌드 패키지를 선택하고, Windows용 Nightly build 패키지를 내려 받는다. 필자가 내려 받은 패키지는 [http://download.berlios.de/codeblocks/CB\\_20060515\\_rev2451\\_win32.7z](http://download.berlios.de/codeblocks/CB_20060515_rev2451_win32.7z) 이다.

◆ 또한 Code::Blocks를 위한 wxWidget DLL을 내려 받는다.

- 내려받을 파일은 [http://download.berlios.de/codeblocks/wxmsw26u\\_gcc\\_cb.7z](http://download.berlios.de/codeblocks/wxmsw26u_gcc_cb.7z) 이다.

◆ 7z 파일은 7-zip 유틸리티로 압축 해제 할 수 있다. 내려받기 주소는 <http://www.7-zip.org> 이다.

◆ 내려받은 CB\_20060515\_rev2451\_win32.7z 파일을 특정 디렉토리에 압축 풀고, 모든 파일을 <IDE\_ROOT>WCodeBlocksW 에 복사하여 기존 파일들을 덮어씌운다.

◆ 내려받은 wxmsw26u\_gcc\_cb.7z 파일을 특정 디렉토리에 압축 풀고, wxmsw26u\_gcc\_cb.dll 파일을 <IDE\_ROOT>WCodeBlocksW 에 복사한다.

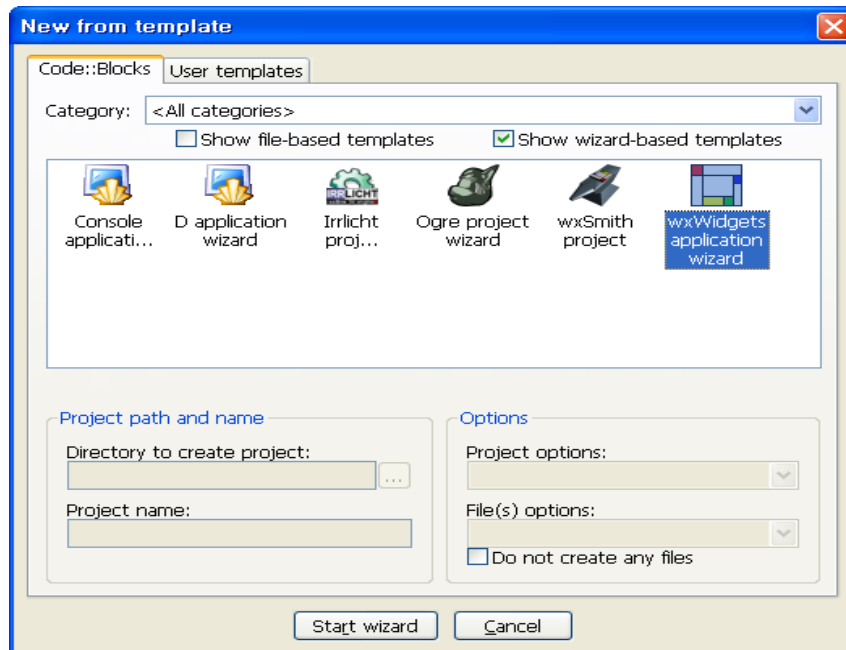
◆ 기존의 <IDE\_ROOT>WCodeBlocksW 에 있는 wxmsw26\_gcc\_cb.dll 는 삭제한다.

### 4. Code::Blocks 설정

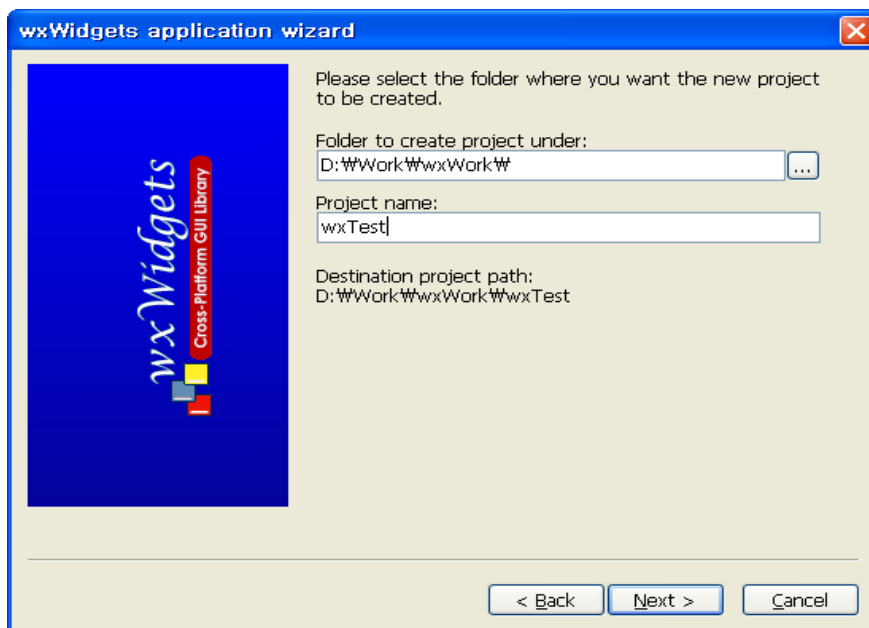
◆ CodeBlocks IDE를 실행한다.

◆ File -> New Project를 실행한다.

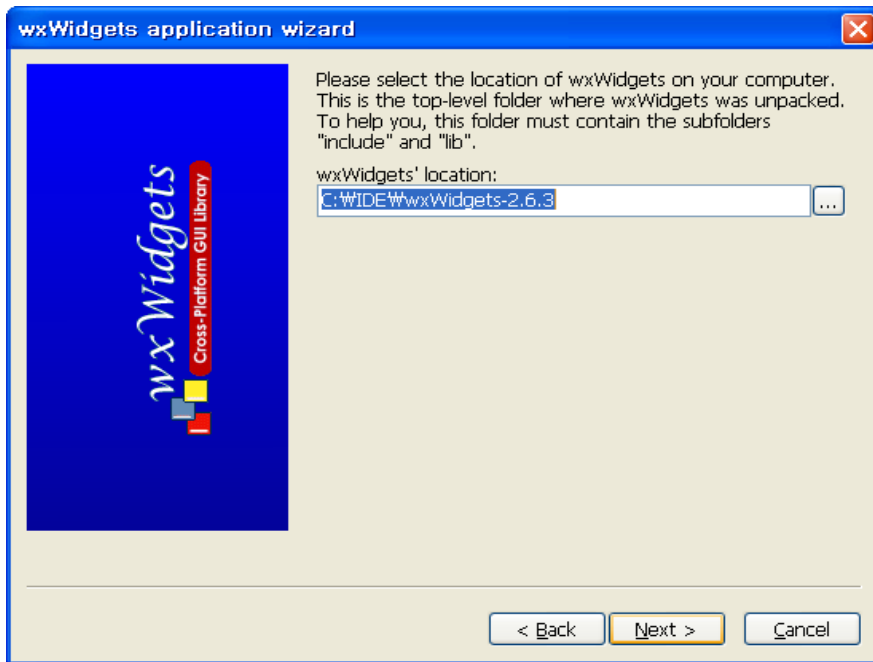
- wxWindgets application wizard를 선택한 후, "start wizard" 를 누른다.



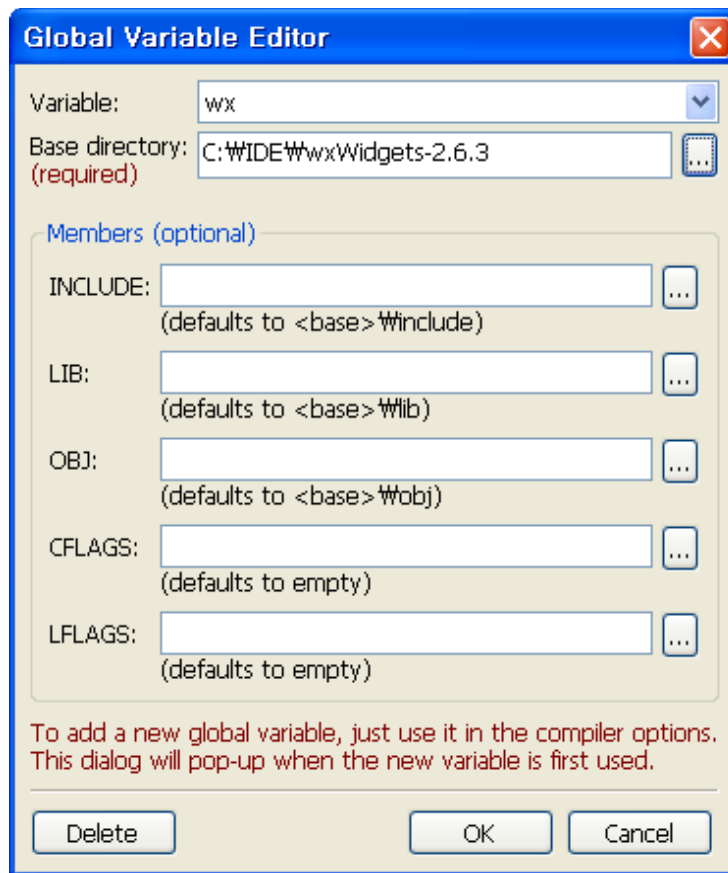
- 프로젝트를 생성할 위치와 프로젝트 이름을 입력한다.



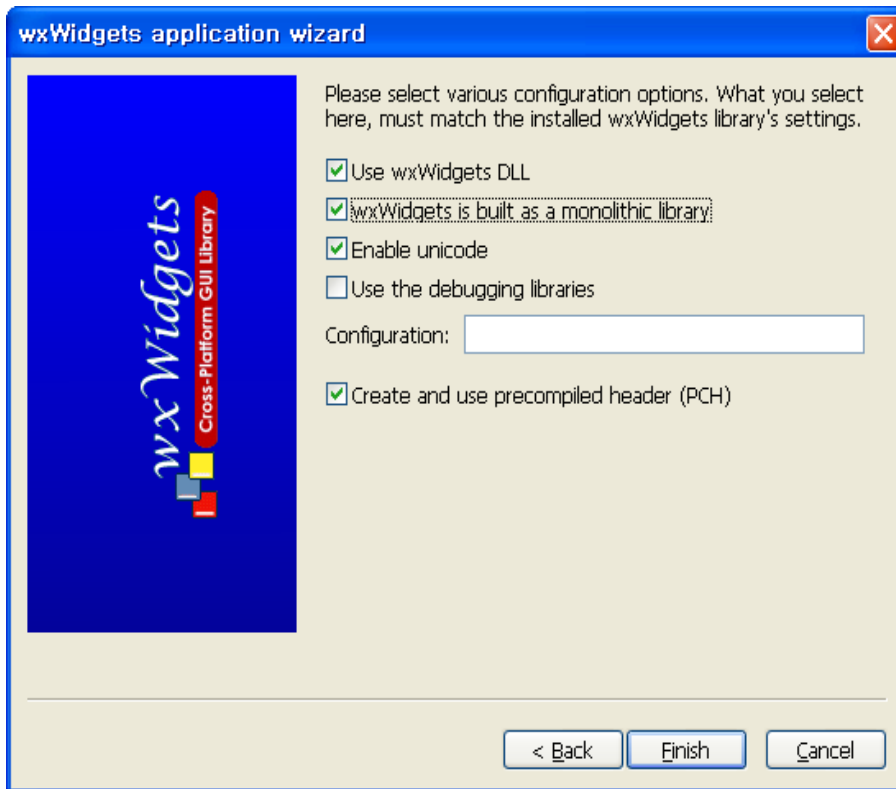
- wxWidgets의 위치를 기술한다.



- Global Variable Editor에 wx Variable에 대한 Base Directory를 입력한다.



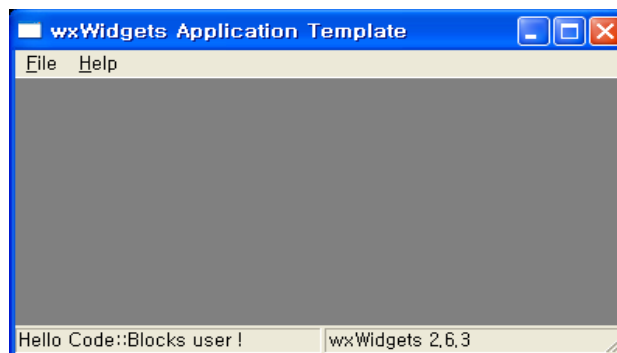
- 마지막 옵션들을 디폴트로 놔두고 "Finish"를 누른다.



- 메뉴에서 "Build"를 해본다. 오류 없이 진행되면 wxTest.exe가 생성된다.

◆ 생성된 실행 프로그램은 실행하기 위하여 **wxmsw26u\_gcc\_custom.dll** 파일이 필요하다. 이 파일은 C:\WIDE\wxWidgets-2.6.3\lib\wgcc\_dll 에 있으며, 이 디렉토리를 PATH 환경 변수에 추가하거나 c:\Windows\System32\ 에 복사한다.

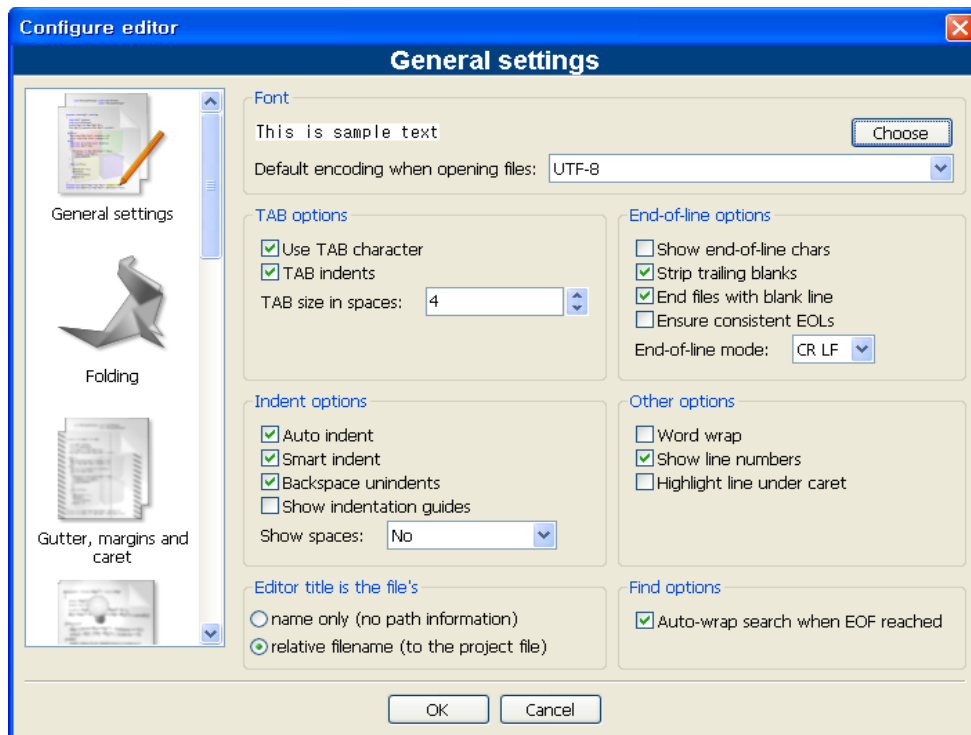
- 프로그램을 실행해 본다. 화면에 나오면 성공이다.



◆ 프로그램이 콘솔창과 함께 뜰 경우, Project/targets options에서 target option을 GUI application으로 바꾸고, "Pause when execution ends"를 꺼준 후, 리빌드 한다.

주:) PATH 환경 변수를 변경할 경우, PATH 를 사용하는 프로그램을 재 실행해 주어야 변경된 환경 변수값을 읽을 수 있다.

◆ 이 글은 UNICODE를 사용하는 것을 전제로 작성하였기 때문에, 에디터 설정 역시 유니코드를 지원하도록 설정하여야 한다.



## 5. 우분투 리눅스에서 구축 방법

◆ Code::Blocks를 설치하기 전에 필요한 패키지들을 설치한다.

```
$ sudo apt-get install build-essential gdb
```

```
$ sudo apt-get install libwxgtk2.6-0 libwxgtk2.6-dev wx2.6-headers wx2.6-doc wx-common
```

◆ <http://forums.codeblocks.org/index.php?board=20.0> 에서 최신의 바이너리 deb 패키지(이 글을 쓰는 시점에서 최신 패키지명은 [http://download.berlios.de/codeblocks/CB\\_20060524\\_rev2499.deb](http://download.berlios.de/codeblocks/CB_20060524_rev2499.deb) 이다)를 다운로드 받는다.

◆ 다운로드 받은 위치에서 패키지를 설치한다.

```
$ sudo dpkg -i <Name_of_Daily_Build.deb>
```

◆ 4. 예서와 같이 Code::Blocks를 설정하기에 앞서 심볼릭 링크를 하나 만들어주어야 한다.

```
$ ln -s /usr/include/wx-2.6/wx /usr/include/wx
```

◆ Global Variable Editor에 wx Variable에 대한 Base Directory를 “/usr”을 입력한다.

◆ wxWidgets Application Wizard를 실행할 때, wxWidgets’s location에 “/usr” 을 입력한다.

◆ 자세한 설치 문서는 [이 곳](http://wiki.codeblocks.org/index.php?title=Installing_Code::Blocks)([http://wiki.codeblocks.org/index.php?title=Installing\\_Code::Blocks](http://wiki.codeblocks.org/index.php?title=Installing_Code::Blocks))에서 찾을 수 있다.

이상으로 MinGW, wxWidget, Code:Blocks를 이용한 C/C++ IDE 환경 구축 및 우분투 리눅스에서의

Code::Blocks 설정 방법을 알아보았다. 정상적으로 설치가 되었으면, [다음 문서](#) ([http://wiki.codeblocks.org/index.php?title=Category:Application\\_Development](http://wiki.codeblocks.org/index.php?title=Category:Application_Development))에서 “WxSmith Tutorial & Pointers”를 직접 따라해 보기 바란다. WxSmith는 GUI 응용프로그램 제작을 돕는 리소스 편집 도구 이다.

필자는 우리나라의 데스크탑 환경이 마이크로소프트 일색인 특정 기업의 독점적 구조를 탈피하기 위해서, MS에 종속적인 개발 환경에서 탈피하여 특정 OS에 독립적인 프로그램들이 많이 나오기를 바란다. 그러기 위해서는 개발 환경이 우선 MS 종속을 탈피해야 한다고 생각한다. 이 글이 어느 정도 필자의 바람을 충족시켜 주기를 기대하면서 글을 마칩니다.

### **참고 사이트**

- <http://www.mingw.org> : Minimalist GNU for Windows
- <http://wxwidgets.org> : Cross-Platform GUI Library
- <http://codeblocks.org> : The Open Source, Cross-platform, Free C/C++ IDE
- <http://wiki.codeblocks.org> : the official Wiki for Code::Blocks