
Developing for wxWidgets using Eclipse and the CDT

Table of Contents

Prerequisites	1
Retrieving the settings	1
Retrieving the settings on Windows	1
Retrieving the settings on Mac OS X	2
Using wxWidgets in eclipse	2
The DLL problem	3
The parser problem	3
Library path problem on Linux	4
Sample Hello World Program	4
Additional notes	5

Note

This document is a so-called one-shot document. It is here to describe the use of wxWidgets with Eclipse for a certain version at a certain time (Spring 2006).

This document is not maintained. I will not answer support questions. I will not make changes or provide additional information.

If you feel like you would like to maintain this document, please let me know and I will provide you with the source files.

Prerequisites

You must have the following items installed:

- Eclipse
- Eclipse CDT, at least version 3.0.2 or newer
- A compiler and build system
- wxWidgets must be installed and compiled

If you do not have these prerequisites, please take a look at my Setting up Eclipse CDT on Windows, Linux/Unix, Mac OS X HowTo [<http://max.berger.name/howto/cdt/>] and my installing wxWidgets HOWTO [<http://max.berger.name/howto/wxWidgets/mingw/>].

Retrieving the settings

Before you can compile wxWidgets programs, you need to know which settings to use.

Retrieving the settings on Windows

To retrieve the settings on windows, open up MinGW (assuming you have followed my MinGW/MSYS instructions. You may have to change things a little for cygwin). Then change to your wxWidgets folder:

```
cd $WXWIN
```

Retrieve the compile settings:

```
./wx-config --cxxflags
```

And the build settings:

```
./wx-config --libs
```

Note: For cut'n'paste just mark the lines with your mouse, they are automatically copied into the clipboard. Just push "Ctrl-V" in another application to paste.

Retrieving the settings on Mac OS X

If you have Mac OS X 10.4 or higher, it comes preinstalled with wxWidgets. Just open up a terminal and type:

```
wx-config --cxxflags
```

for the compile settings and

```
wx-config --libs
```

for the build settings. In OS X, mark your text with the mouse, then use Apple-C to copy it to the clipboard, and in your target application use Apple-V to paste it.

Using wxWidgets in eclipse

To create a wxWidgets project in eclipse, first create a new "Managed make C++ project" as usual. Then

- Make sure the main project folder is selected in the C/C++ Projects view.
- Select Project / Properties from the menu.
- Select C/C++ Build.
- Under Configuration, make sure it shows the configuration you want to work with. In most cases, this is "debug". If you change that to "Release" then you have to do these things over again.
- Make sure the "Tool Settings" tab is selected.
- Under GCC C++ Compiler, select "Miscellaneous"
- Where it says "other flags" there is already a "-c -fmessage-length=0". After that, add a space and add all the compile settings (use cut'n'paste!).
- Under GCC C++ Linker, select "Miscellaneous"
- Where it says "Linker flags" add all the build setting flags.
- Go to GCC C++ Linker, directly.
- In the Box where it says "Expert Settings: Command Line Pattern" you will have something like:

```
 ${COMMAND} ${FLAGS} ${OUTPUT_FLAG}${OUTPUT_PREFIX}${OUTPUT} ${INPUTS}
```

If you don't have the expert settings box, then your CDT version is too old. Please update to at least 3.0.2 or newer.

You need to change that to have `${FLAGS}` as the last in the list, like this:

```
 ${COMMAND} ${OUTPUT_FLAG}${OUTPUT_PREFIX}${OUTPUT} ${INPUTS} ${FLAGS}
```

Please make sure you add spaces where they should be and don't add spaces where they should not be.

- On Mac OS X you will also have to create an application bundle.

Switch to the tab "Build Steps"

For Post-Build Step, add the following (in one line and with the ; in the middle):

```
mkdir -p ${BuildArtifactFileName}.app/Contents/MacOS ;  
mv ${BuildArtifactFileName} ${BuildArtifactFileName}.app/Contents/MacOS/
```

After you compile, use Finder to execute your new application

- On some Mac OS X versions (PowerPC) there is a bug in g++ 4.0 (noted in 4.0.1, build 5243), which is the default compiler. If you get strange linker errors, try the following:
 - under GCC C++ Compiler set Command to g++-3.3 (instead of g++)
 - under MacOS X C++ Linker set Command to g++-3.3 (instead of g++)
 - make sure you recompile everything (Project / Clean, then rebuild)
- On Mac OS X with XCode 2.2 the g++ compiler (gcc 4.0.1 build 5243) has a bug that prevents it from linking. In this case, under "GCC C++ Compiler" and "GCC C++ Linker" you will need to set the command to "g++-3.3" to use the older (bug free) version of gcc.
- Click "Ok". Your project is now configured for wxWidgets.

The DLL problem

If you have compiled wxWidgets to be "shared" and you are on windows (you have not followed my wxWidgets compilation instructions [<http://max.berger.name/howto/wxWidgets/mingw/>]), then all your programs will need certain DLLs from the wxWidgets to run.

- Find the .dll files (most likely in \$WXWIN/build or \$WXWIN/lib)
- Copy these to the same location as your .exe (for example your debug folder in your project)

The parser problem

If you've followed the instructions given here, you will most likely get a Yellow Questionmark on your `#include "wx/wx.h"` line. This is because we've bypassed some of the configuration.

You can open up the project settings, and go to "GCC C++ Compiler" / "Directories". Here, add all the include paths (the ones starting with -I in the compile settings) but remove the -I !

Library path problem on Linux

On Linux, you may have to add `/usr/local/lib` to your `LD_LIBRARY_PATH`:

In Run -> Debug... , add "`LD_LIBRARY_PATH=/usr/local/lib`" in the Environment tab. Repeat for Run-> Run...

Sample Hello World Program

Create a new source file in your project, and copy the following code in there:

```
/*
 * hworld.cpp
 * Hello world sample by Robert Roebeling
 * Adapted for unicode by Max Berger
 */

#include "wx/wx.h"

class MyApp: public wxApp
{
    virtual bool OnInit();
};

class MyFrame: public wxFrame
{
public:

    MyFrame(const wxString& title,
            const wxPoint& pos, const wxSize& size);

    void OnQuit(wxCommandEvent& event);
    void OnAbout(wxCommandEvent& event);

    DECLARE_EVENT_TABLE()
};

enum
{
    ID_Quit = 1,
    ID_About,
};

BEGIN_EVENT_TABLE(MyFrame, wxFrame)
    EVT_MENU(ID_Quit, MyFrame::OnQuit)
    EVT_MENU(ID_About, MyFrame::OnAbout)
END_EVENT_TABLE()

IMPLEMENT_APP(MyApp)
```

```
bool MyApp::OnInit()
{
    MyFrame *frame = new MyFrame( wxT("Hello World"),
        wxPoint(50,50), wxSize(450,340) );
    frame->Show(TRUE);
    SetTopWindow(frame);
    return TRUE;
}

MyFrame::MyFrame(const wxString& title,
    const wxPoint& pos, const wxSize& size)
: wxFrame((wxFrame *)NULL, -1, title, pos, size)
{
    wxMenu *menuFile = new wxMenu;
    menuFile->Append( ID_About, wxT("&About...") );
    menuFile->AppendSeparator();
    menuFile->Append( ID_Quit, wxT("E&xit") );

    wxMenuBar *menuBar = new wxMenuBar;
    menuBar->Append( menuFile, wxT("&File") );

    SetMenuBar( menuBar );

    CreateStatusBar();
    SetStatusText( wxT("Welcome to wxWindows!") );
}

void MyFrame::OnQuit(wxCommandEvent& WXUNUSED(event))
{
    Close(TRUE);
}

void MyFrame::OnAbout(wxCommandEvent& WXUNUSED(event))
{
    wxMessageBox(wxT("This is a wxWindows Hello world sample"),
        wxT("About Hello World"), wxOK | wxICON_INFORMATION, this);
}
```

Then compile and run! If you've followed these instructions it should work!

Additional notes

Thanks to Andrew Challen:

Thanks for helping me setup wxWidgets with Eclipse in Windows and Ubuntu I did this in Ubuntu first, then Windows XP which was much harder but I found and fixed a problem not covered in your notes, detailed below if you are interested. [...]

after getting 'compile settings' with wx-config --cxxflags in mingw the paths need modifying to C:\ style instead of msys /c/ style and with 'build settings' e.g.

```
-I/c/wxWidgets-2.8.10/lib/wx/include/msw-unicode-release-static-2.8 -I/c/wxWidgets  
-L/c/wxWidgets-2.8.10/lib -mthreads -Wl,--subsystem,windows -mwindows /c/wxWidge  
to  
-Ic:\wxWidgets-2.8.10\lib\wx\include\msw-unicode-release-static-2.8 -Ic:\wxWidgets  
-Lc:\wxWidgets-2.8.10\lib -mthreads -Wl,--subsystem,windows -mwindows c:\wxWidge
```

Also, and this had me stuck for ages my program would not link properly claiming it couldn't find many libraries to solve:

```
in Eclipse->Project->Properties->C/C++ Build->Settings->Tool Settings->MingW C+  
+ Linker->Miscellaneous->Expert settings Change from ${COMMAND} ${FLAGS}  
${OUTPUT_FLAG}${OUTPUT_PREFIX}${OUTPUT} ${INPUTS} to ${COMMAND} ${INPUTS}  
${FLAGS} ${OUTPUT_FLAG}${OUTPUT_PREFIX}${OUTPUT}
```

`\${INPUTS}` at the end simply didn't work![...]