

Tutorial – gtkmm unter Windows

Installation

GTK+ und gtkmm Pakete installieren.

Verwendete Pakete:

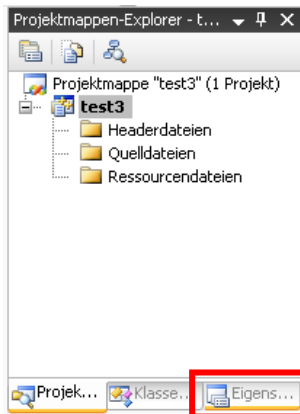
- gtk-dev-2.12.9-win32-2.exe
- gtkmm-win32-devel-2.12.7-3.exe

Diese Anleitung funktioniert vermutlich auch mit anderen Versionen.

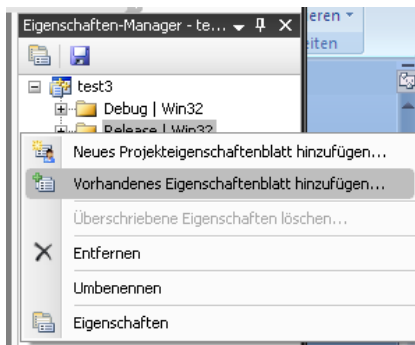
1) Kompilieren mit Visual Studio 2008 (VC 9)

Hinweis: gtkmm funktioniert bei mir mit dieser Vorgehensweise nur mit der *Release*-Konfiguration.

1. Neues Projekt anlegen > Visual C++ > win32 > Win32-Konsolenanwendung.
2. Anwendungseinstellungen > „Leeres Projekt“ auswählen.
Wie und ob es mit vorkompilierten Headern funktioniert, habe ich nicht getestet.
3. Dann auf Eigenschaften-Manager (auch unter *Ansicht* > *weitere Fenster* zu finden).



4. Dann auf Rechtsklick auf *Release* und auf *Vorhandenes Eigenschaftenblatt hinzufügen*.



5. {GTK-PATH}\MSVC\gtkmm-2.4.vspops auswählen. GTK-PATH ist der Installationspfad von GTK, normalerweise C:\GTK.
6. Quelldatei hinzufügen und Beispielcode einfügen:

```
#include <gtkmm.h>

int main(int argc, char *argv[])
{
```

Autor: Marc Schlaich (marc.schlaich [at] web.de)

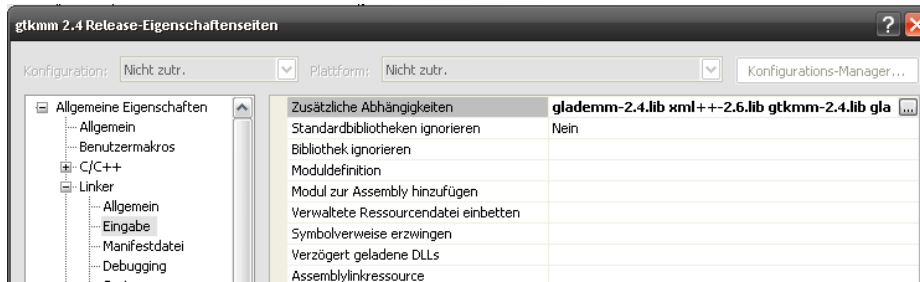
Datum: 17. Februar 2009

```

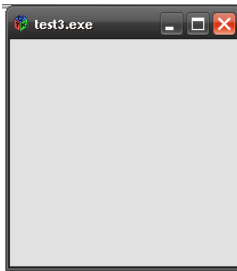
    Gtk::Main kit(argc, argv);
    Gtk::Window window;
    Gtk::Main::run(window);
    return 0;
}

```

7. Konfiguration auf *Release* stellen und *Debugging starten*.
Bei einem Compiler-Fehler („gionmm-2.4.lib wurde nicht gefunden“) folgende Schritte ausführen:
8. Wechsel auf *Eigenschaften-Manager* und Doppelklick auf „gtkmm 2.4 Release“.
9. Erweitern zu *Linker > Eingabe*.



10. *Zusätzliche Abhängigkeiten* über den `...`-Button editieren.
11. Entsprechenden Eintrag („gionmm-2.4.lib“) suchen und diese Zeile löschen.
12. Nun müsste über *Debugging starten* das gtkmm-Fenster erscheinen:



2) Kompilieren mit MinGW

MinGW Installation

MinGW mit Hilfe des *Automated Installer* installieren, dabei mindestens die Pakete „gcc“ und „g++“ auswählen.

Batchfile erstellen

Nun eine Batchdatei „g++2.bat“ unter *MinGW/bin* mit folgendem Inhalt erstellen:

```

@echo off
pkg-config gtkmm-2.4 --cflags --libs > gtk-flags
set /p flags=<gtk-flags
g++ %* %flags%

```

Nun kann der Befehl „g++2“ wie der normale „g++“ Befehl genutzt werden, nur das zusätzlich noch die GTK-Flags, die über den Aufruf von pkg-config erzeugt werden, als Parameter übergeben werden. Der Aufruf „g++2 [Parameter]“ ist äquivalent zum Linux-Aufruf: „g++ [Parameter] `pkg-config gtkmm-2.4 --cflags --libs`“.

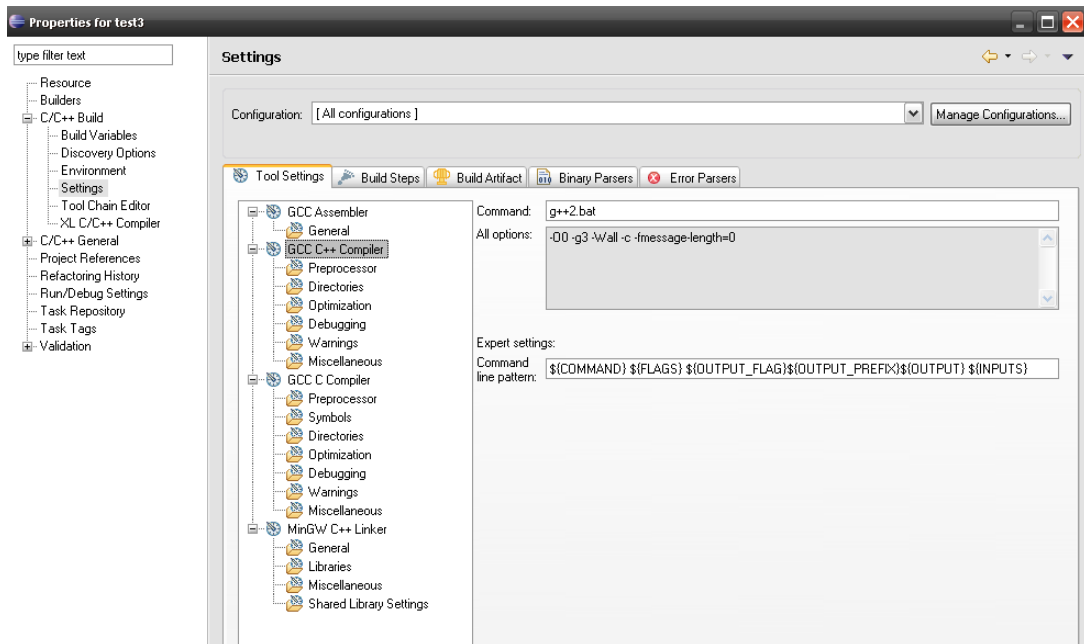
Autor: Marc Schlaich (marc.schlaich [at] web.de)

Datum: 17. Februar 2009

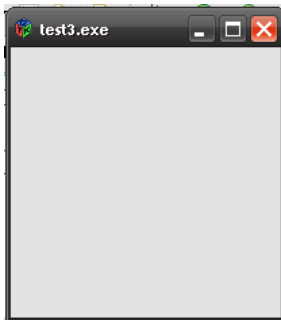
3) Kompilieren mit Eclipse CDT

Voraussetzung: Schritte in Teil 2 „Kompilieren mit MinGW“ durchgeführt.

1. C++ Projekt mit MinGW als Compiler anlegen.
2. Quelldatei mit Beispielcode anlegen (Bsp. s. Teil 1)
3. Nun auf *Project > Properties*.
4. Erweitern auf *C/C++ Build > Settings*.
5. Unter *Configurations: „[All configurations]“* auswählen.
6. Nun unter *GCC C++ Compiler > Command: „g++2.bat“* eintragen.



7. Abschließend unter *MinGW C++ Linker > Command: „g++2.bat“* eintragen.
8. Mit F11 (Debug Run) und dann F8 (Resume) taucht auch hier das gtkmm-Window auf:



©Copyright – Bitte nur im Original veröffentlichen.