

G++ unter Windows

mit MinGW

*Präsentation von Patrick Dammann
für IPK SoSe'15 und IPI WiSe'15/16*

Download

- Google: MinGW
 - erster Eintrag
 - ([MinGW | Minimalist GNU for Windows](#))
 - oben rechts Download-Button
- oder direkt unter:
 - <http://sourceforge.net/projects/mingw/>

Download von MinGW

[About](#) [Downloads](#) [Documentation](#) [Recent posts](#) [Feed aggregator](#)

MinGW
Minimalist GNU for Windows

Search this site:



Welcome to MinGW.org Home of the MinGW and MSYS Projects

MinGW, a contraction of "Minimalist GNU for Windows", is a minimalist development environment for native Microsoft Windows applications.

MinGW provides a complete Open Source programming tool set which is suitable for the development of native MS-Windows applications, and which do not depend on any 3rd-party C-Runtime DLLs. (It **does** depend on a number of DLLs provided by Microsoft themselves, as components of the operating system; most notable among these is MSVCRT.DLL, the Microsoft C runtime library. Additionally, threaded applications **must** ship with a freely distributable thread support DLL, provided as part of MinGW itself).

MinGW compilers provide access to the functionality of the Microsoft C runtime and some language-specific runtimes. **MinGW**, being Minimalist, **does not, and never will, attempt to provide a POSIX runtime environment for POSIX application deployment on MS-Windows**. If you want POSIX application deployment on this platform, please consider [Cygwin](#) instead.

Primarily intended for use by developers working on the native MS-Windows platform, but also available for cross-hosted use, (see note below -- you may need to follow the "[read more](#)" link to see it), **MinGW** includes:

- A port of the GNU Compiler Collection (GCC), including C, C++, ADA and Fortran compilers;
- GNU Binutils for Windows (assembler, linker, archive manager)
- A command-line installer, with optional GUI front-end, (mingw-get) for MinGW and MSYS deployment on MS-Windows
- A GUI first-time setup tool (mingw-get-setup), to get you up and running with mingw-get.

MSYS, a contraction of "Minimal SYStem", is a Bourne Shell command line interpreter system. Offered as an alternative to Microsoft's cmd.exe, this provides a general purpose command line environment, which is particularly suited to use with **MinGW**, for porting of many Open Source applications to the MS-Windows platform; a light-weight fork of Cygwin-1.3, it includes a small selection of Unix tools, chosen to facilitate that objective.

[Read more](#) 522336 reads

[Download Installer](#)

Search mingw users

Choose Mail List:

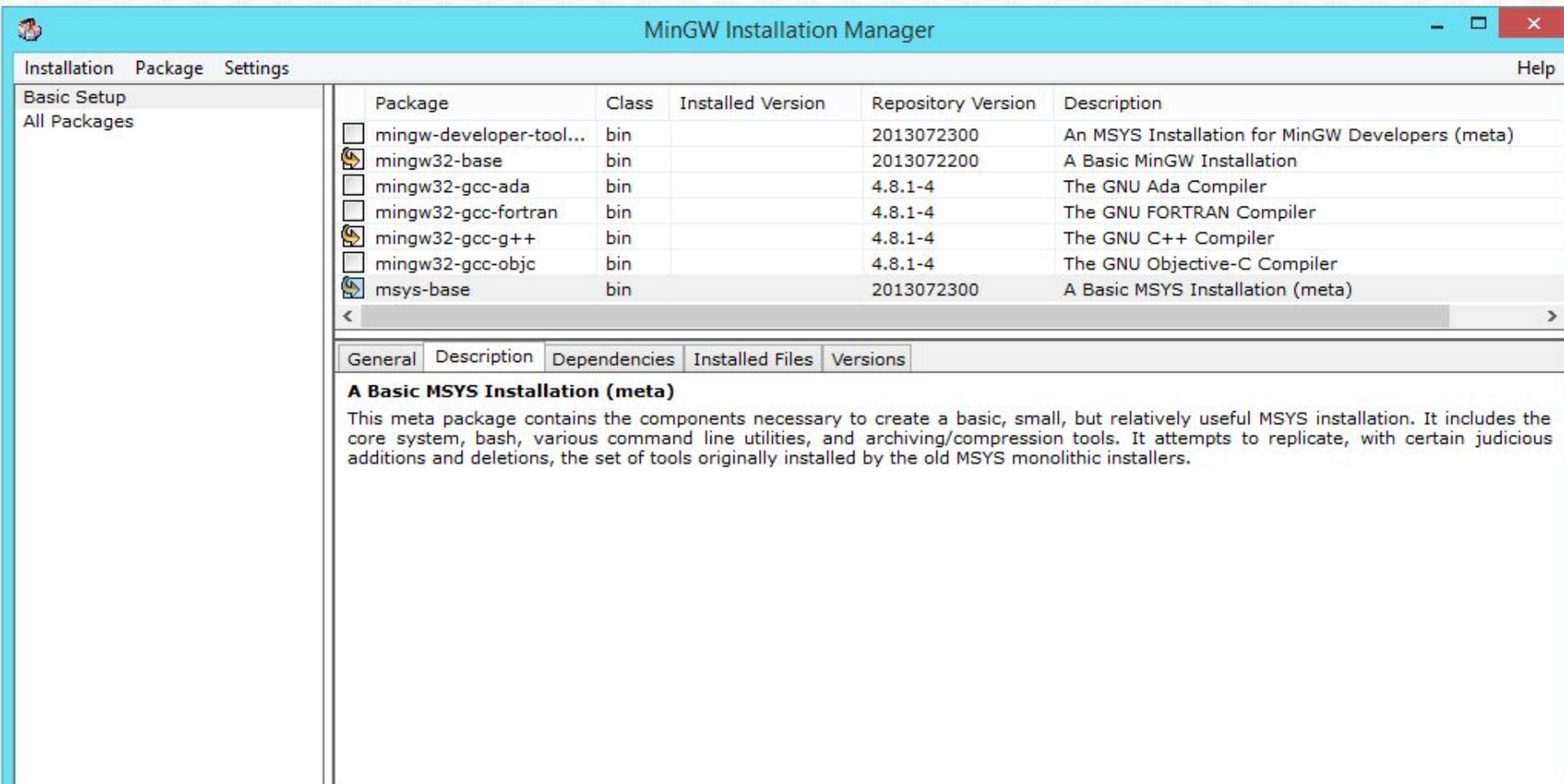
MinGW-users ▼

Recent File Releases

- /MSYS/Base/bash/bash-3.1.23-1/bash-3.1.23-1-msys-1.0.18.README.txt
- /Installer/mingw-get/catalogue/msys-bash.xml.lzma
- /MSYS/Base/bash/bash-3.1.23-1/bash-3.1.23-1-msys-1.0.18-bin.tar.xz
- /MSYS/Base/bash/bash-3.1.23-1/bash-3.1.23-1-msys-1.0.18-doc.tar.xz
- /MSYS/Base/bash/bash-3.1.23-1/bash-3.1.23-1-msys-1.0.18-lic.tar.xz
- /MSYS/Base/bash/bash-3.1.23-1/bash-3.1.23-1-

Installation MinGW

Intuitiv bis zum Ende “durchklicken”.



The screenshot shows the MinGW Installation Manager window. The title bar reads "MinGW Installation Manager". The menu bar includes "Installation", "Package", "Settings", and "Help". The left sidebar shows "Basic Setup" and "All Packages". The main area displays a table of packages:

Package	Class	Installed Version	Repository Version	Description
<input type="checkbox"/> mingw-developer-tool...	bin		2013072300	An MSYS Installation for MinGW Developers (meta)
<input checked="" type="checkbox"/> mingw32-base	bin		2013072200	A Basic MinGW Installation
<input type="checkbox"/> mingw32-gcc-ada	bin		4.8.1-4	The GNU Ada Compiler
<input type="checkbox"/> mingw32-gcc-fortran	bin		4.8.1-4	The GNU FORTRAN Compiler
<input checked="" type="checkbox"/> mingw32-gcc-g++	bin		4.8.1-4	The GNU C++ Compiler
<input type="checkbox"/> mingw32-gcc-objc	bin		4.8.1-4	The GNU Objective-C Compiler
<input checked="" type="checkbox"/> msys-base	bin		2013072300	A Basic MSYS Installation (meta)

Below the table, the "General" tab is selected, showing the description for "A Basic MSYS Installation (meta)":

A Basic MSYS Installation (meta)
This meta package contains the components necessary to create a basic, small, but relatively useful MSYS installation. It includes the core system, bash, various command line utilities, and archiving/compression tools. It attempts to replicate, with certain judicious additions and deletions, the set of tools originally installed by the old MSYS monolithic installers.

PATH-Variable anpassen

- Rechtsklick auf Computer
- Eigenschaften
- Links: Erweiterte Systemeinstellungen
- Reiter: Erweitert
- Unten: Umgebungsvariablen
- Unteres Feld: “Path” suchen
- Bearbeiten
- “;C:\MinGW\lib;C:\MinGW\bin;” ANHÄNGEN
(!)
- NICHT ersetzen

Das hier gilt für **Win 7**, **Win 8** und **Win 8.1**.

Für alles andere einfach mal nach “path variable ändern Win XP/Win Vista/Win 10/Win 95SE” googlen.

System

Systemsteuerung > System

Startseite der Systemsteuerung

Basisinformationen über den Computer anzeigen

Geräte-Manager

Remoteeinstellungen

Computerschutz

Erweiterte Systemeinstellungen

Windows-Edition

Windows 8.1

© 2013 Microsoft

Systemeigenschaften

name Hardware Erweitert Computerschutz Remote

Systemvariable bearbeiten

Name der Variablen: Path

Wert der Variablen: %PATH%;C:\Program Files\Broadcom\Broadcom 802.11n\bin;C:\MinGW\bin;

OK Abbrechen

Systemvariablen

Variable	Wert
NUMBER_OF_P...	4
OS	Windows_NT
Path	C:\Program Files\Broadcom\Broadcom 802.11n\bin;C:\MinGW\bin;
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;...

Neu... **Bearbeiten...** Löschen

OK Abbrechen

Umgebungsvariablen...

Notepad++

- Download unter <http://notepad-plus-plus.org/>
 - (kann man natürlich auch googlen)
- Installation sollte intuitiv sein
- Dazu das Plug-In “nppexec”
 - Download unter: <http://goo.gl/pmFcCZ>
- Installation von nppExec
 - zip-File entpacken
 - dll in C:\Programme\Notepad++\Plugins kopieren

NppExec konfigurieren

- Via “Erweiterungen->NppExec->Execute”
- 2 Scripte anlegen
- Ein Kompilier-Script:

```
NPP_SAVE
```

```
CD "$ (CURRENT_DIRECTORY) "
```

```
g++.exe -std=c++11 $ (FILE_NAME) "
```

```
-o "$ (NAME_PART) .exe "
```

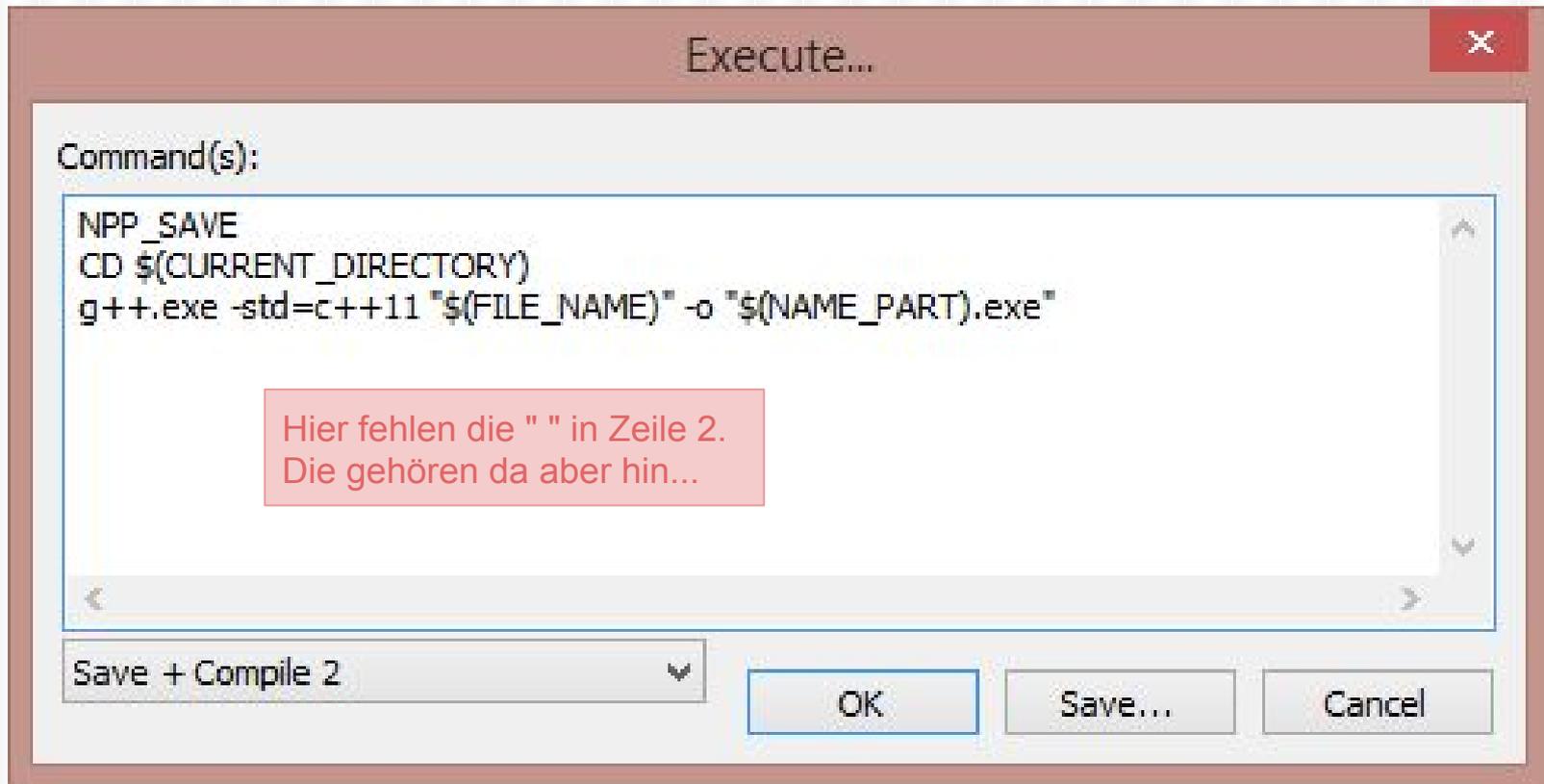
- Ein Ausführ-Script:

```
CD "$ (CURRENT_DIRECTORY) "
```

```
"$ (NAME_PART) .exe "
```

NppExec konfigurieren

.....



Erweiterungen-> NppExec-> Advanced Options

Haken setzen NppExec Advanced Options

Menu items *

Place to the Macros submenu

G++ Compile :: G++ Compile
G++ Exec :: G++ EXEC

Move up Move down

* Assign shortcut keys using Settings:
Shortcut Mapper -> Plugin commands

Menu item

Item name:
G++Exec

Associated script:
G++ EXEC

Add/Modify Delete

Advanced options

Execute this script when Notepad++ starts:
[Dropdown]

Execute this script when Notepad++ exits:
[Dropdown]

[Options]

HotKey: F6 [Dropdown]
ToolbarBtn: Console [Dropdown]

[Console]

Visible (on start): Auto [Dropdown]
ShowHelp (on start): No [Dropdown]
SaveCmdHistory: Yes [Dropdown]
CommentDelimiter: // [Text]
TextColorNormal (RR GG BB): FF FF FF [Text]
TextColorError (RR GG BB): A0 10 10 [Text]
TextColorMessage (RR GG BB): 11 F5 3F [Text]
BackgroundColor (RR GG BB): 00 00 00 [Text]

OK Cancel

click
(für beide Scripte)

Shortcuts

Über *Einstellungen* -> *Tastatur* -> *Plug-In Commands* lassen sich jetzt Shortcuts anlegen (z.B. Shift+F7 zum kompilieren und Shift+F8 zum Ausführen)

Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Makro Ausführen Erweiterungen

Fenster ?

X



hw.cpp

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Hello World!" << endl;
6 }
```

Console

X

```
"C:\Users\S...ent\Documents\C++2014\hw.cpp.exe
Process started >>>
Hello World!
<<< Process finished. (Exit code 0)
===== READY =====
```



length: 88 lines: 6 Ln: 6 Col: 2 Sel: 0|0

UNIX

ANSI as UTF-8

INS

...