

1 MSys / MinGW Information

This document describes the development environments **MSYS** and **MinGW**. These packages provide an environment using the GNU compiler collection (**gcc**) to build typical Open Source projects like Tcl/Tk under Windows.

1.1 Short Description

1.1.1 MSYS

Short description from the homepage of MSYS:

<http://www.mingw.org/>

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.

MSYS is a collection of Unix tools for Windows. It contains all tools which are needed for the typical build process using the `configure / make` toolset.

Examples: `autogen, cp, rm, mv, mkdir, m4, make`

MSYS is available as 32-bit version only. This version can be used in conjunction with both the 32-bit and 64-bit version of **MinGW**.

1.1.2 MinGW

Short description from the homepage of MinGW-w64:

<http://sourceforge.net/projects/mingw-w64/>

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.

MinGW provides the GNU Compiler Collection **gcc** for Windows. The SourceForge project **MinGW-w64** supplies 32-bit and 64-bit versions of **gcc**.

The **MinGW-w64** project also supplies an extended version of **MSYS** (see chapter Installation below for details).

1.2 Installation (Short story)

Download MSYS/MinGW ready-to-use packages:

- 32-bit: [gcc4.9.2_i686-w64-mingw32.7z](#)
- 64-bit: [gcc4.9.2_x86_64-w64-mingw32.7z](#)

Extract the Zip file(s) on drive C:.

Start the MSYS Shell by double-clicking onto file

```
C:\gcc4.9.2_i686-w64-mingw32\msys\msys.bat      or  
C:\gcc4.9.2_x86_64-w64-mingw32\msys\msys.bat
```

If you extracted the files into another directory, change the line in file `YourPath\msys\etc\fstab` to fit your path.

```
C:/gcc4.9.2_i686-w64-mingw32/mingw32      /mingw
```

Note:

- Use forward slashes
- Don't use pathes with spaces

2 Installation (Long story)

2.1.1 Download MSYS

Entry page:

<http://sourceforge.net/projects/mingwbuilds/files/external-binary-packages/>

File:

msys+7za+wget+svn+git+mercurial+cvs-rev13.7z

Link:

<http://sourceforge.net/projects/mingwbuilds/files/external-binary-packages/msys%2B7za%2Bwget%2Bsvn%2Bgit%2Bmercurial%2Bcvs-rev13.7z/download>

2.1.2 Download MinGW

Entry page for 32-bit version:

<http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/>

Entry page for 64-bit version:

<http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Personal%20Builds/mingw-builds/>

32-bit gcc 4.9.2

File:

i686-4.9.2-release-posix-dwarf-rt_v4-rev4.7z

Link:

http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/4.9.2/threads-posix/dwarf/i686-4.9.2-release-posix-dwarf-rt_v4-rev4.7z/download

32-bit gcc 5.2.0

File:

i686-5.2.0-release-posix-dwarf-rt_v4-rev0.7z

Link:

http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/5.2.0/threads-posix/dwarf/i686-5.2.0-release-posix-dwarf-rt_v4-rev0.7z/download

32-bit gcc 7.2.0

File:

i686-7.2.0-release-posix-dwarf-rt_v5-rev1.7z

Link:

https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/7.2.0/threads-posix/dwarf/i686-7.2.0-release-posix-dwarf-rt_v5-rev1.7z/download

64-bit gcc 4.9.2

File:

x86_64-4.9.2-release-posix-seh-rt_v4-rev4.7z

Link:

http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Personal%20Builds/mingw-builds/4.9.2/threads-posix/seh/x86_64-4.9.2-release-posix-seh-rt_v4-rev4.7z/download

64-bit gcc 5.2.0

File:

x86_64-5.2.0-release-posix-seh-rt_v4-rev0.7z

Link:

http://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Personal%20Builds/mingw-builds/5.2.0/threads-posix/seh/x86_64-5.2.0-release-posix-seh-rt_v4-rev0.7z/download

64-bit gcc 7.2.0

File:

x86_64-7.2.0-release-posix-seh-rt_v5-rev1.7z

Link:

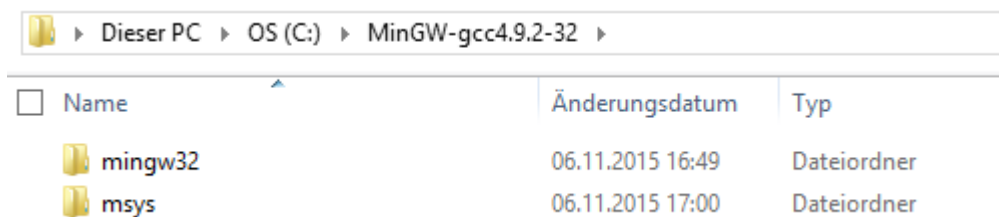
https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Personal%20Builds/mingw-builds/7.2.0/threads-posix/seh/x86_64-7.2.0-release-posix-seh-rt_v5-rev1.7z/download

2.1.3 Extract

The following instructions use the 32-bit version of gcc 4.9.2. Installation is done on drive C:. Adapt file and directory names accordingly, when using other versions.

- Create directory `C:\MinGW-gcc4.9.2-32`
- Extract downloaded MinGW file in directory `C:\MinGW-gcc4.9.2-32`
- Extract downloaded MSYS file in directory `C:\MinGW-gcc4.9.2-32`

Your directory structure should now look as follows:



Name	Änderungsdatum	Typ
mingw32	06.11.2015 16:49	Dateiordner
msys	06.11.2015 17:00	Dateiordner

2.1.4 Configuration

Insert the next two lines into file `C:\MinGW-gcc4.9.2-32\msys\etc\fstab`

```
# Win32_Path          Mount_Point
C:/MinGW-gcc4.9.2-32/mingw32  /mingw
```

2.1.5 Test

Start the MSYS Shell by double-clicking onto file `C:\MinGW-gcc4.9.2-32\msys\msys.bat`. You may create a shortcut of `msys.bat` on your desktop for easier access.

2.1.6 Compile Tcl and Tk

- Use the **configure** script contained in the `win` subdirectory.
- Use the following **configure** options for 32-bit builds:
 - `--build=x86_64-w64-mingw32`
- Use the following **configure** options for 64-bit builds:
 - `--build= x86_64-w64-mingw32 --enable-64bit`

2.2 Further informations

Source: <http://sourceforge.net/p/mingw-w64/wiki2/MSYS/>

Where to get MSYS

There are three places you can get MSYS:

- The [MinGW project](#), with separate packages of all official MSYS packages. Takes a long time to download and install everything.
- The all-in-one package on the [MinGW-w64 download page](#). It is updated on request (see third option for very up to date collection)
- [MinGW-builds](#) provides an ultra-inclusive MSYS package with a bunch of additional useful stuff.

How to use the MSYS

Installing MSYS is quite easy.

- You'll need to download the above package.
- Unzip it somewhere, for example `C:\msys` so that `C:\msys\bin` contains (among others) `bash.exe`.
- Doubleclick (or make a handy shortcut and run that) on `C:\msys\msys.bat`.
- Type `sh /postinstall/pi.sh`
- Answer the friendly questions and you're all set up.

Mingw-w64/w32 specifics

When running an autotools configure script, these options will come in handy:

- for a 64-bit build: `--host=x86_64-w64-mingw32`
- for a 32-bit build: `--host=i686-w64-mingw32`

If you are experiencing problems, you can also set `--build` to the same value. Some configure scripts also use `--target` instead of `--host`. Use `configure --help` to get all possible options.

--host, --target, and --build explained

`--host` specifies on what platform/architecture the compiled program is going to run. `--target` specifies the platform/architecture that the program should be configured for and will be compiled for. This should only have effect when building cross-compilers. `--build` specifies the platform/architecture the build process is going to be executed.

What is MSYS

MSYS is a Minimal SYStem, providing several crucial unix utilities under a compatibility layer (the `msys-1.0.dll` file). MSYS should provide everything to make compilation of common GNU software. An outdated [description](#) by the makers themselves.

MSYS provided by the mingw-w64/w32 project

This package is not more than a collection of the 50+ packages provided by mingw.org. It was created as a (huge) convenience to our users, to let them be productive instead of downloading every part separately. The accompanying sources are also provided and can be found in the same download section as mentioned above.

This package is 32-bit, but will run flawlessly on x64 Windows. There will never be a 64-bit native MSYS (is there any need?) because the only compiler capable of building MSYS applications is the outdated gcc 3.4.4, which does not support x64 native Windows targets.