

Setup MinGW

MSYS2

Download & Install

Download MSYS2 from:

<https://msys2.github.io/>

Install in e.g. c:/msys

Configure

Update the core first:

```
update-core
```

Then restart MSYS2 Shell and update the remaining packages:

```
pacman -Su
```

MinGW Toolchain

Package-Install summary:

```
pacman -S mingw-w64-i686-gcc mingw-w64-x86_64-gcc make mingw-w64-i686-libffi  
mingw-w64-x86_64-libffi zip
```

Compiler

Install using MSYS pacman:

32-bit:

```
pacman -S mingw-w64-i686-gcc
```

64-bit:

```
pacman -S mingw-w64-x86_64-gcc
```

Build tools

Make:

```
pacman -S make
```

MinGW-Make (not needed):

```
pacman -S mingw-w64-i686-make mingw-w64-x86_64-make
```

Make command is called „mingw32-make“ (on 64-bit too) in „c:/msys/mingw32/bin“.

Libraries

Install additional packages using pacman. Search for packages using:

```
pacman -Ss package_name_of_something_i_want_to_install
```

```
pacman -S mingw-w64-i686-libffi mingw-w64-x86_64-libffi
```

The headers (“ffi.h” among others) are located in “c:/msys/mingw32/lib/libffi-3.2.1/include”. Copy them to e.g. “c:/msys/mingw32/include”, so they are found by gcc (the same applies for mingw64).

Tools

Perl:

```
pacman -S perl
```

Install URI::Escape module (requires „make“):

```
$ perl -MCPAN -e shell
cpan[1]> install URI::Escape
cpan[2]> q
```

ZIP (for make bindist) and unzip (for nsis-installer):

```
pacman -S zip
pacman -S unzip
```

7-zip (for make bindist):

```
pacman -S p7zip
```

A rather old version is in the repository. Rather consider to use a version installed in “C:/Program Files/7-Zip” by adding that directory to PATH.

Dos2unix (for nsis-installer):

```
pacman -S dos2unix
```

The package contains the tools “dos2unix” as well as “unix2dos”; only the latter is needed by the nsis-installer makefile.

Environment

fstab

The mounted directories can be specified in the fstab file located in “c:/msys/etc/fstab”. An entry consists of <Windows Path> followed by <MSYS Path> and separated by spaces or tabs.

Unfortunately, spaces in Paths are not supported, so old 8.3 names have to be used as a workaround. Use “dir /x” command to list the 8.3 names.

A fstab may contain these (additional to the default cygdrive one) entries for example:

```
C:/fb_dev    /fb_dev
# paths with spaces not supported (!)
# "C:/Program Files (x86)/FreeBASIC"
```

```
C:/PROGRA~2/FreeBASIC /fb
```

In case no 8.3 names are available or other problems occur with the restriction of spaces in paths the directories can be mounted manually using the mount command alternatively. One can put that in the profile initialization (see below).

The “fstab.d” directory seems to be without any function – at least creating “00_freebasic.fstab” file there was ignored. So the “fstab” file has to be edited directly. See

<http://askubuntu.com/questions/168290/why-cant-mount-read-files-in-etc-fstab-d>

profile

The profile contains the environment initialization code (like setting the PATH variable). To avoid modifying the profile file directly, which is located in “c:/msys/etc/profile”, rather create an initialization script in the “profile.d” subdirectory located in the same folder.

For example, create a shell script “c:/msys/etc/profile.d/freebasic.sh” with the following content to add paths to the PATH variable:

```
#mounting done in fstab
#mount -f 'C:/Program Files (x86)/FreeBASIC' /fb
PATH="/c/Program Files/7-Zip:${PATH}"
PATH="/c/Program Files (x86)/NSIS:${PATH}"
PATH="/fb:${PATH}"
```

It also contains a sample to mount a path with spaces, which is supported by the mount command.