

MSYS2 and You: Easier Windows Native Dependencies

Zakariyya Mughal

2020 June 25

TPC 2020: Conference in the Cloud



<https://github.com/zmughal/talk-tpc20cic-msys2-lightning-talk>

Problem

- P0 You have Perl code that needs to run on Windows (e.g., user requirements, Windows-only software/hardware support).
- P1 It can't be done in pure-Perl.
- P2 It requires many tricky-to-install native dependencies.

Problem

- P0 You have Perl code that needs to run on Windows (e.g., user requirements, Windows-only software/hardware support).
- P1 It can't be done in pure-Perl.
- P2 It requires many tricky-to-install native dependencies.

Problem

- P0 You have Perl code that needs to run on Windows (e.g., user requirements, Windows-only software/hardware support).
- P1 It can't be done in pure-Perl.
- P2 It requires many tricky-to-install native dependencies.

Terminology

- Cygwin: POSIX-like environment for Windows (uses `cygwin1.dll`).
- MinGW: GCC-based toolchain for Windows (32/64-bit)
- MSYS: part of the MinGW project; shell and utilities forked from Cygwin a long time ago (e.g., run `./configure`).
- MSYS2: project to combine MinGW with updates from Cygwin and a package manager.
- Other approaches: Windows Subsystem for Linux.

Terminology

- Cygwin: POSIX-like environment for Windows (uses `cygwin1.dll`).
- MinGW: GCC-based toolchain for Windows (32/64-bit)
- MSYS: part of the MinGW project; shell and utilities forked from Cygwin a long time ago (e.g., run `./configure`).
- MSYS2: project to combine MinGW with updates from Cygwin and a package manager.
- Other approaches: Windows Subsystem for Linux.

Terminology

- Cygwin: POSIX-like environment for Windows (uses `cygwin1.dll`).
- MinGW: GCC-based toolchain for Windows (32/64-bit)
- MSYS: part of the MinGW project; shell and utilities forked from Cygwin a long time ago (e.g., run `./configure`).
- MSYS2: project to combine MinGW with updates from Cygwin and a package manager.
- Other approaches: Windows Subsystem for Linux.

Terminology

- Cygwin: POSIX-like environment for Windows (uses `cygwin1.dll`).
- MinGW: GCC-based toolchain for Windows (32/64-bit)
- MSYS: part of the MinGW project; shell and utilities forked from Cygwin a long time ago (e.g., run `./configure`).
- MSYS2: project to combine MinGW with updates from Cygwin and a package manager.
- Other approaches: Windows Subsystem for Linux.

Terminology

- Cygwin: POSIX-like environment for Windows (uses `cygwin1.dll`).
- MinGW: GCC-based toolchain for Windows (32/64-bit)
- MSYS: part of the MinGW project; shell and utilities forked from Cygwin a long time ago (e.g., run `./configure`).
- MSYS2: project to combine MinGW with updates from Cygwin and a package manager.
- Other approaches: Windows Subsystem for Linux.

Who uses MSYS2

- Rtools (starting with R 4.0.0),
- RubyInstaller,
- PyGObject, ...
- and several other FOSS projects recommend it as their build environment.

Installing All Your Base

- Go to <https://www.msys2.org/> and follow the installation instructions (or use the Chocolatey to automate).

- Open a terminal for one of the subsystems:

MSYS For building code that requires the compatibility layer.

MINGW32 For 32-bit code.

MINGW64 For 64-bit code. This is most likely what you want.

- Install Perl for MINGW64 and the toolchain (for compiling XS)

```
pacman -S mingw-w64-x86_64-perl mingw-w64-x86_64-toolchain
```

Installing All Your Base

- Go to <https://www.msys2.org/> and follow the installation instructions (or use the Chocolatey to automate).
- Open a terminal for one of the subsystems:

MSYS For building code that requires the compatibility layer.

MINGW32 For 32-bit code.

MINGW64 For 64-bit code. This is most likely what you want.

- Install Perl for MINGW64 and the toolchain (for compiling XS)

```
pacman -S mingw-w64-x86_64-perl mingw-w64-x86_64-toolchain
```

Installing All Your Base

- Go to <https://www.msys2.org/> and follow the installation instructions (or use the Chocolatey to automate).
- Open a terminal for one of the subsystems:

MSYS For building code that requires the compatibility layer.

MINGW32 For 32-bit code.

MINGW64 For 64-bit code. This is most likely what you want.

- Install Perl for MINGW64 and the toolchain (for compiling XS)

```
pacman -S mingw-w64-x86_64-perl mingw-w64-x86_64-toolchain
```

Install your packages...for great justice.

- Find your package by either browsing <https://github.com/msys2/MINGW-packages> or searching at the prompt:

```
pacman -Ss <query>
```

Examples: SDL2, FFTW, GStreamer.

- Install the packages

```
pacman -S mingw-w64-x86_64-gstreamer \  
mingw-w64-x86_64-gst-plugins-base \  
mingw-w64-x86_64-gst-plugins-good \  
mingw-w64-x86_64-gobject-introspection
```

Install your packages...for great justice.

- Find your package by either browsing <https://github.com/msys2/MINGW-packages> or searching at the prompt:

```
pacman -Ss <query>
```

Examples: SDL2, FFTW, GStreamer.

- Install the packages

```
pacman -S mingw-w64-x86_64-gstreamer \  
mingw-w64-x86_64-gst-plugins-base \  
mingw-w64-x86_64-gst-plugins-good \  
mingw-w64-x86_64-gobject-introspection
```

Further information

- Steps on building Gtk3.pm <http://project-renard.github.io/doc/development/meeting-log/posts/2016/05/03/windows-build-with-msys2/>.
- Script that automates installation of Gtk3.pm: <https://github.com/MaxPerl/perl-Gtk3-windows-installer>.

Questions / Contact

Reach out

- on IRC: sivoais on `irc://irc.perl.org/#native` (we can also chat about Alien and FFI!),
- on Twitter: @zmughal,
- on GitHub: <https://github.com/zmughal>.