Building SWI-Prolog for MS-Windows using MinGW

The MinGW compiler suite is a port of GCC that targets the Windows platform. Unlike Cygwin which comes with an extensive POSIX runtime emulation, MinGW targets the native Windows API including MSVCRT (The MSVC runtime library that provides some POSIX capabilities).

MinGW is distributed as a cross-compiler for many Linux distributions and can be used under MSYS, "a collection of GNU utilities such as bash, make, gawk and grep to allow building of applications and programs which depend on traditionally UNIX tools to be present. It is intended to supplement MinGW and the deficiencies of the cmd shell.".

The details for building SWI-Prolog for MS-Windows under Linux are documented in the executable readme file `README.mingw` of the source distribution. These scripts have been developed on 64-bit Ubuntu 12.04 and are currently used for building the binary releases on Ubuntu 16.04.

Building on Windows itself

According to Matthias Gondon the core system (i.e., without packages) can be build on Windows as follows:

- Download mingw-get from http://www.mingw.org, run the program
- From the “Basic setup”, install packages `mingw32-developer-toolkit, mingw32-base, mingw32-gcc, mingw32-gfortran, mingw32-g++`
- From “All packages”, select `mingw32-gmp`
- Extract current swipl sources to a folder that **does not contain spaces**
- `cd swipl-devel/src`
- `autoconf`
- `autoheader`
- `./configure`
- `make`

This requires a version **after Feb 6, 2017** (GIT, releases 7.4.0-rc2 and 7.5.1)

Building the full system is probably possible using the instructions in the above mentioned `README.mingw` file. Please send comment and or fixes if you try this. Note that cross compiling is much faster because SWI-Prolog depends on an extensive `configure` script that executes slow on the MSYS bash implementation.

See also

- Daily builds