swi-prolog.org

SWI-Prolog -- Installation on Linux, *BSD (Unix)

5-6 minutes

Installing from binaries

The official packages are often out of date. PPAs are created automatically for every release.

PPAs are available for Ubuntu by following directions on the PPA page.

Installing from source

Installing from source is often the best option for installing on Linux and *BSD based systems. Building is not complex.

Downloading SWI-Prolog

Download the SWI-Prolog source

- As a *tar ball* from the download page
- Using GIT from GIThub

Preparing the source (when downloading using GIT)

The *tar ball* is self contained. The GIT repository contains sub modules and does not contain generated files. The GIT source is downloaded and prepared for building using the commands below:

```
git clone https://github.com/SWI-Prolog/swipl-devel.git
cd swipl-devel
./prepare
```

Similarly, the sequence to update the source using git and rebuild is

```
make distclean
git pull
./prepare
./build
```

Getting the prerequisites

Building SWI-Prolog from source requires tools as well as libraries. We maintain pages that describes the required dependencies by platform. The last entry of the list below describes the dependencies in platform independent way.

- Debian based linux (Debian, Ubuntu, Mint, Raspberry Pi, ...)
- Redhat based linux (Fedora, Redhat, CentOS, ...)
- SuSE
- Mageia

- Other systems (background info)

Using the build.templ script

Both to facilitate frequent installs and to show some common options, you are invited to use the **build.templ** script located in the top directory. Copy this file to **build** and edit it according to the [EDIT] instructions in the script. Then simply run the script as below. This script is fairly independent from versions, so preserve your customised script for later releases.

```
% tar zxvf swipl-<version>.tar.gz
% cd swipl-<version>
% cp -p build.templ build
% <edit> build
% ./build
```

If the above fails, follow the instructions below.

Doing it in multiple steps

The configuration on Unix is based on GNU-Autoconf, a set of scripts that automatically extract system characteristics and creates the Makefile and C config header config.h. The generated Makefile only works with GNU-Make. The installation requires an ANSI-C compiler. Configure prefers the GNU-project C-compiler gcc, but many other compilers work fine too. The first step, building SWI-Prolog lite, is outlined below.

```
% gzip -d < swipl-<version>.tar.gz | tar xvfB
```

```
-
% cd swipl-<version>/src
% ./configure                    # see bellow
% make
% [su root]                      # for public
installation
% make install
```

The second step, building all tools, is only supported for machines on which building and loading shared libraries is supported, luckily this is the case on almost all modern operating systems.

The packages are built and installed from the packages directory in the main distribution directory. The file README there contains additional information on building the packages. The basic procedure is:

```
% cd swipl-<version>/packages
% ./configure
% make
% [su root]                      # for public
installation
% make install
```

Finally, packages can be installed one-by-one. With pl installed in a your $PATH, simply go to the desired package directory and run the standard GNU/Linux installation procedure. Global configure options such as installation location are extracted from Prolog. Some package configure

scripts provide additional options that specify the location of required libraries.

```
% ./configure
% make
% [sudo] make install
```

configure is a Bourne Shell script doing the actual configuration for your computer. The most important options of confure are:

--help
: Print a list of all available options. See the files INSTALL and INSTALL.notes in the distribution for additional information.

--prefix=prefix
: Set the base directory for the installation. The executables are stored in prefix/bin/, the library in prefix/lib /pl-%plversion/ and the include files in prefix/include. For public installation (by root), /usr/local is the default. For private installation it is normal to specify $HOME

--exec_prefix=prefix
: Overrides the directory for installing the executables to exec_prefix/bin/, normally used on heterogeneous networks.

After succesfull installation, the compilation directory may be removed.

In case of trouble ...

The file INSTALL in the source directory contains the generic documentation about installing autoconf-based programs. The file INSTALL.notes in the same directory contains additional SWI-Prolog related hints.

If `configure` fails, try using `bash configure`. If `make` fails, try using GNU-make and gcc.

If all fails, have a look at the Support page.

Post installation (JPL)

If you want to call Java from Prolog using JPL, you need to add the directory holding the JVM shared objects to the dynamic linker search path. Using default installation on Ubuntu, this is achieved by adding the following to your `~/.profile`:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/jvm
/java-1.8.0-openjdk-amd64/jre/lib/amd64/server
```