

Installation

By default, the **FITSH** package does not have any specific dependence. You only have a C compiler (e.g. GCC ^[1]) and the related standard development headers (that are usually placed in the directory `/usr/include`) on your system. See also the list of supported operating systems and architectures below.

Here we detail how **FITSH** can be installed:

- system-wide,
- for a single user, or
- using a deb(ian) based package manager.

In principle, **FITSH** do not need any special privileges on the host computer, so all of the features are instantly available for a non-privileged normal user. System-wide and package-based installation simplifies the access to **FITSH** tasks for all of the users having an access to the computer in general. In this description below, the commands beginning with the prompt `# . . .` refer to commands needed to be issued by the root (i.e. system administrator) while commands beginning with the prompt `$. . .` can be run without any specific privileges.

System-wide installation

The current stable version of the program is **<version/>**, you can browse the available versions here ^[2]. Installing the package from the source tarball might go as follows. After extracting the archive, by entering simply (for the latest version):

- `cd /usr/src`
- `wget http://fitsh.net/download/fitsh/fitsh-<version/>.tar.gz`
- `tar xvzf fitsh-<version/>.tar.gz`

Then, go to the appropriate sub-directory (**fitsh-<version/>**), by issuing `cd fitsh-<version/>`, and run the standard `configure && make` procedure, i.e.:

- `cd fitsh-<version/>`
- `./configure`

which is followed by

- `make`

After a successful compilation, the binaries and some related libraries might be installed into the target filesystem hierarchy location by entering

- `make install`

The "root" of the filesystem hierarchy is the `/usr/local` directory by default. In order to place the binaries to another location, use an alternative **prefix** for `./configure`, for instance:

- `./configure --prefix=/opt`
-

Local installation for a single user

If you do not have rights on the computer for system administration, i.e. you cannot install anything under `/opt` or `/usr/local`, you can safely use your home directory (or a subdirectory in it) in order to install the package. Supposing that your username is `myuser` (see also the output of the `whoami` command) and your home directory is `/home/myuser`, simply type

- `./configure --prefix=/home/myuser/usr`

The installation procedure (see `make install` above) will then create the subdirectories `/home/myuser/usr/bin`, `/home/myuser/usr/lib` and `/home/myuser/usr/include` where the task binaries, `lfit` shared objects and `lfit` module headers are installed, respectively. In this case, you should put the directory `/home/myuser/usr/bin` into your path in order to access the FITSH tasks from everywhere (at least, from your account). Follow your shell's guidelines to do so, if you use the bash shell, put something similar into your `~/.bashrc` file:

```
export PATH=/home/myuser/usr/bin:${PATH}
```

Install as a Debian package

If the DPKG packaging system ^[3] and the `help2man` ^[4] utility is installed on your system, you can use the target

- `make deb`

to create a Debian package. In order to install this newly created package, use the command `dpkg` ^[3]:

- `dpkg --install ./fitsh_<version/>_amd64.deb`

or use your alternative/favourite Debian package manager. Note that the newly created Debian package is always placed into the root of the source tree. Note also that while the `make deb` target doesn't require root privileges, the `dpkg --install ...` does so.

Supported operating systems

During the development of the FITSH package, it was tested under the following operating systems and architectures:

- Debian GNU/Linux ^[5], specifically releases lenny, squeeze, wheezy and jessie. All of these are tested on i386 and amd64 architectures. This is also the primary development platform of the package.
- NetBSD ^[6], release 5.0.2 on i386 architecture.
- Mac OS X ^[7], release 10.6.8 ("Snow Leopard").

The compiler was always the system default C compiler from the GNU Compiler Collection (GCC ^[1]). Since the primary development platform is Debian, some compilation-time warnings may occur on other operating systems.

References

[1] <http://gcc.gnu.org/>

[2] <http://fitsh.net/release>

[3] <http://packages.debian.org/stable/dpkg>

[4] <http://www.gnu.org/software/help2man/>

[5] <http://www.debian.org/>

[6] <http://www.netbsd.org/>

[7] <http://www.apple.com/macosx/>