MutekH is a flexible kernel, with most configuration tweaks resolved at compile time. Therefore, it can only be distributed as source. There is no library building involved.

Somes pages of this wiki refer to specific versions of the MutekH source tree. This is intended. Here, we'll provide general installation notes, therefore, we'll never use an explicit revision.

Prerequisites to build MutekH

For a Linux host (as a process)

In order to compile MutekH as a process, like described in QuickStartUnix, you only need a working GCC and the header files for your libc. Please refer to your distribution manual to know more about this.

For SoCLib and other embedded platforms

For embedded platforms, you'll need:

- a working cross-compiler for your target architecture; you may either follow ?soclib:InstallationNotes, or use tools/crossgen.mk, see below
- `dtc`, an utility to handle device trees, available at ?http://git.jdl.com/software/.

Getting MutekH

This is a simple svn checkout. Pay attention there are some subversion clients that cannot handle ?svn:externals feature. In doubt, a vanilla svn client is recommended.

```
svn co https://www.mutekh.org/svn/mutekh/trunk/mutekh
```

If you need an explicit revision, simply add `-r number` to the command line:

```
svn co -r 1234 https://www.mutekh.org/svn/mutekh/trunk/mutekh
```

Building cross-compilers

SoCLib installation nodes already provides a way to get cross-compilers, if you don't already have them, MutekH holds a tool to build a complete cross-compilation toolchain:

The script is in tools/crossgen.mk.

There is an inline help:

```
 $ tools/crossgen.mk
[prints some help]
```

You can try a line like this one to get a Mips cross-compiler:

```
 $ tools/crossgen.mk all TARGET=mipsel-unknown-elf
```