McStas installation instructions

Peter Willendrup,
peter.willendrup@risoe.dk

May 20, 2003

Abstract

This document describes installation of the McStas package, including some information on installing other required pieces of software.

The text below is also included as a chapter in the McStas manual.

1 Licensing

The conditions on the use of McStas can be read in the files `LICENSE` and `LICENSE.LIB` in the distribution. Essentially, McStas may be used and modified freely, but copies of the McStas source code may not be distributed to others. New or modified component and instrument files may, however, be shared by the user community.

2 Getting McStas

The McStas package is available in three different distribution packages, from the project website at http://mcstas.risoe.dk, e.g.

- `mcstas-1.7-src.tar.gz`
  Source code package for building McStas on (at least) Linux and Windows 2000. This package should compile on most Unix platforms with an `ANSI-c` compiler. - Refer to section 3
- `mcstas-1.7-i686-unknown-Linux.tar.gz`
  Binary package for Linux systems, currently built on Debian GNU/Linux 3.0 'woody'. Should work on most Linux setups. - Refer to section 4
- `mcstas-1.7-i686-unknown-Win32.zip`
  Binary package for Win32 systems, currently built on Microsoft

Windows 2000 professional, using the `gcc` 2.95 compiler from Bloodshed Dev-C++ 5 Beta 7 - Refer to section 5

3 Source code build

The McStas package is beeing co-developed for mainly Linux and Windows systems, however the Linux build instructions below will work on most Unix systems. For an updated list of platforms on which McStas has been built, refer to the project website.

3.1 Windows build

- Start by unpacking the `mcstas-1.7-src.tar.gz` package using e.g. Winzip.
- Using an `ansi-c` compiler (we recommend Bloodshed Dev-C++ - easy to install and use (Section 6.1)), the McStas package can be compiled using the `build.bat` script of the `mcstas-1.7` directory you just unpacked. Follow the on screen instructions.
- When the build has been done (e.g. `mcstas.exe` has been produced), proceed to install (Section 5).

3.2 Unix build

McStas uses autoconf to detect the system configuration and create the proper Makefiles needed for compilation. On Unix-like systems, you should be able to compile and install McStas using the following steps:

1. Unpack the sources to somewhere convenient and change to the source directory:
   gunzip -c mcstas-1.7-src.tar.gz — tar xf -
   cd mcstas-1.7/
2. Configure and compile McStas:
   ./configure
   make
3. Install McStas:
   make install

You should now be able to use McStas. For some examples to try, see the examples/ directory.
The installation of McStas in step 3 by default installs in the /usr/local/ directory, which on most systems requires superuser (root) privileges. To install in another directory, use the –prefix= option to configure

in step 2. For example,

./configure –prefix=/home/joe

will install the McStas programs in /home/joe/bin/ and the library files needed by McStas in /home/joe/lib/mcstas/.

In case ./configure makes an incorrect guess, some environment variables can be set to override the defaults:

- The CC environment variable may be set to the name of the C compiler to use (this must be an ANSI C compiler). This will also be used for the automatic compilation of McStas simulations in mcgui and mcrun.
- CFLAGS may be set to any options needed by the compiler (eg. for optimization or ANSI C conformance). Also used by mcgui/mcrun.
- PERL may be set to the path of the Perl interpreter to use.

To use these options, set the variables before running ./configure. Eg.

```
setenv PERL /pub/bin/perl5
./configure
```

It may be necessary to remove configure's cache of old choices first:

```
rm -f config.cache
```

If you experience any problems, or have some questions or ideas concerning McStas, please contact peter.willendup@risoe.dk or the McStas mailing list at neutron-mc@risoe.dk.

You should try to make sure that the directory containing the McStas binaries (mcstas, gscan, mcdisplay, etc.) is contained in the PATH environment variable. The default directory is /usr/local/bin, which is usually, but not always, included in PATH. Alternatively, you can reference the McStas programs using the full path name, ie.

```
/usr/local/bin/mcstas my.instr
perl /usr/local/bin/mcrun -N10 -n1e5 mysim -f output ARG=42
perl /usr/local/bin/mcdisplay --multi mysim ARG=42
```

This may also be necessary for the front-end programs if the install procedure could not determine the location of the perl interpreter on your system.

If McStas is installed properly, it should be able to find the files it needs automatically. If not, you should set the MCSTAS environment variable to the directory containing the runtime files "mcstas-r.c" and "mcstas-r.h" and the standard components (*.comp). Use one of

```
MCSTAS=/usr/local/lib/mcstas; export MCSTAS # sh, bash
setenv MCSTAS /usr/local/lib/mcstas         # csh, tcsh
```

To get a fully working McStas environment, you must also install an `ANSI-c` compiler, for instance `gcc`. A prebuilt package probably exists for your system. To get a functional graphical user interface and plotting facilities, you must also install the following packages:

- Perl, Tk and perl-Tk for GUI (see Section 6.2)
- Matlab, Scilab or (PGPLOT+pgperl+PDL) for plotting (see Section 6.3)

4 Binary install, Linux

Should be very easy, simply start from 'make install' in Section 3.

5 Binary install, Windows

- Start by unpacking the `mcstas-1.7-i686-unknown-Win32.zip` package using e.g. Winzip.
- Execute the `install.bat` installation script. Follow the on screen instructions.
- Set the required (see output of `install.bat`) environment variables using

  'Start/Settings/Control Panel/System/Advanced/Environment Variables'

- To get a fully working McStas environment, you must also install an `ansi-c` compiler, for instance BloodShed Dev-C++ (Section 6.1)
- To get a functional graphical user interface and plotting facilities, you must also install the following packages:
  - ActivePerl and ActiveTcl for GUI (see Section 6.2)
  - Matlab or Scilab for plotting (see Section 6.3)

6 Installing support Apps

6.1 Bloodshed Dev-C++ (Win32)

To install Bloodshed Dev-C++, download the installer package from

http://www.bloodshed.net/dev/devcpp.html.

When installed, add the `Dev-Cpp\bin` directory to your path using

'Start/Settings/Control Panel/System/Advanced/Environment Variables'.

6.2 Gui tools (Perl + Tk) (All platforms)

- Win32:
  - Get and install the ActivePerl package from http://www.activestate.com/Products/Download/Register.plex?id=ActivePerl (Registration not required)
  - Get and install the ActiveTcl package from http://www.activestate.com/Products/Download/Register.plex?id=ActiveTcl (Registration not required)
- Unix:
  - Install Perl, Tk and perl-Tk. Prebuilt packages exist for most Linux distributions and also most other Unix-like operating systems.
  - Consult the McStas webpage at http://mcstas.risoe.dk for updated links to the source code distributions.

6.3 Plotting backends (All platforms)

For plotting with McStas, different support packages can be used:

- PGPLOT/PDL/pgperl (Unix only) - Binary builds of the packages exist for various Linux distributions (for instance Debian comes with prebuilt versions). Prebuilt versions also exist for some commercial Unix'es. Refer to distributor/vendor for documentation. The packages can also be built from source using some (in many cases much) effort. See the PGPLOT documentation for further details.
- Matlab (Some Unix/Win32) - refer to http://www.mathworks.com. Matlab licenses are rather costly, but discount programmes for university and research departments exist.

- Scilab (Unix/Win32/Mac...) - a free 'Matlab-like' package, available from http://www-rocq.inria.fr/scilab/. McStas also requires the Plotlib library from http://www.dma.utc.fr/~mottelet/myplot.html.

7 Testing the McStas distribution

The `examples` directory of the distribution contains an automatic distribution test procedure, which is executed (under Unix-like systems) with:

```
./examples/selftest
```

This test takes a few minutes to complete, and should essentially display messages saying that some of the components used in the test instruments are obsolete (but still work). Additionally, plots from results may be generated automatically from the command

```
./examples/showtest
```

which will use by default the PGPLOT plotting back-end (see below). Other back-ends (see Section 6.3) may be specified when setting the MCSTAS_FORMAT environment variable.