[Ubuntu] [Apple OS X] [xspec local models] [installation test I]

by M. Nowak

The following is a description of the steps I took to build a set of X-ray data analysis software on my laptop running the linux operating system, Ubuntu 9.04. My specific set-up consists of a MacBook Pro (running Mac OS X, version 10.5.7), with the linux partition being run via a virtual machine (VMWare, version 2.0.4 (159196)). Note that some editing of files in the "vmware-tools -distrib " directory on the Ubuntu partition had to be done in order for the VMware tools (which allow one to drag and drop files between the Mac and linux partitions, as well as use the Mac hard disk from the linux partition) to run properly on Ubuntu 9.04. (See http://communities.vmware.com/thread/206772)

The reason for building the tools on the linux partition is that the Apple operating system does not handle dynamically loaded libraries as easily or smoothly as a real linux system. Although the tools can be built on Mac OS X 10.4 or 10.5 (few analysis systems continue to support 10.3, especially on PPC architectures), the builds are more difficult, and later expansion (especially with "local models" incorporated via the XSPEC analysis program) can be tricky.

As can be seen below, the build with a linux system is relatively straightforward. The whole process should take under two hours. (If it is taking longer, you probably have problems, and should ask for help in determining what's going wrong!) Here I outline the steps.

1) First we build Ubuntu 9.04 (instructions not given!), making sure that we are up-to-date with all of the packages listed below. The packages can be obtained and built via " apt-get install". For example,

```
unix%> apt-get install build-essential
```

Here are the required packages:

```
build-essential
libreadline5-dev
libncurses5
libncurses5-dev
libpng12-0
libpng12-dev
x-dev
libx11-dev
libxt-dev
```

gfortran
libcurl3
libcurl3-dev
libgtk2.0-dev
libpcre3
libpcre3-dev

Note that here and throughout that one likely needs "root" permissions to install these packages, as well as to build all of the software listed below. Alternatively, one can instead use the unix "sudo" command to invoke the proper permissions to build these packages. For example,

unix%> sudo apt-get install build-essential

will work, if the sudo command recognizes the user account as being allowed to invoke it with the proper permissions.

We will assume throughout that such steps are being taken.

2) Note that above we are using the gfortran compiler for building Fortran libraries. In general, it is best to build all the programs with a consistent set of compilers. It is much more likely that the resultant libraries will interact smoothly with one another. Here are the versions of the C, C++, and Fortran compilers that I am using:

unix%> gcc --version
gcc (Ubuntu 4.3.3-5ubuntu4) 4.3.3

unix%> g++ --version
g++ (Ubuntu 4.3.3-5ubuntu4) 4.3.3

unix%> gfortran --version
GNU Fortran (Ubuntu 4.3.3-5ubuntu4) 4.3.3

3) We will first build the HEASOFT suite of software. As of June 1, 2009, the current version is HEASOFT 6.6.2. One can download the source code (as a compressed tar file) from:

http://heasarc.gsfc.nasa.gov/lheasoft/download.html

The download page allows the user to choose subsets of the software packages. This can save on download time, build time, and disk space. As many people will later on want to use these tools to extract spacecraft data, as well as analyze data, it is more convenient to choose the "all" option, and build the entire HEASOFT 6.6.2 release.

For my machine, I chose "all" software, and downloaded the package to:

/usr/local/heasoft6.6.2src.tar.gz

One can place and build the software in any locations (and not necessarily the same locations for the download and builds); however, much of the software below will first search "standard" locations below the "/usr/local" directories. Thus, for ease of builds, throughout we will default to the " /usr/local" directory structure for both the downloads and the builds.

4) We now follow the standard instructions for installing HEASOFT, as described on the HEASOFT installation page:

http://heasarc.nasa.gov/lheasoft/install.html

First, we uncompress the package:

```
unix%> tar zxvf heasoft6.6.2src.tar.gz
```

Next, we want to make sure that we are using the compilers listed above. Here the syntax we use presumes that we are working in a csh variant (as opposed to a bash variant, for example).

```
unix%> setenv CC /usr/bin/gcc
unix%> setenv CXX /usr/bin/g++
unix%> setenv FC /usr/bin/gfortran
```

We then follow the usual HEASOFT install instructions, again following the syntax for using a cs
h
variant (the assumption we adopt for all the instructions given below).

```
unix%> cd /usr/local/heasoft-6.6.2/BUILD_DIR/
unix%> ./configure >& config.out &
unix%> make >& build.log &
unix%> make install >& install.log &
```

The above steps may take over an hour on a typical laptop.

For purposes of initiating the HEASOFT tools, one needs to set the HEADAS environment variable:

```
unix%> setenv HEADAS /usr/local/heasoft-6.6.2/i686-pc-linux-gnu-libc2.9/
```

The "i686-pc-linux-gnu-libc2.9" sub-directory was created in the build process. On other architectures, it might have a slightly different name.

HEASOFT then can be initialized via:

```
unix%> source $HEADAS/headas-init.csh
```

5) Next, we build the S-lang scripting language, with the current version as of June 1, 2009 being S-lang 2.1.4. We obtain the source code download from www.s-lang.org, and unpack it to:

```
/usr/local/slang-2.1.4
```

Building S-lang is then very straightforward:

```
unix%> cd /usr/local/slang-2.1.4
unix%> ./configure --with-readline=gnu
unix%> make
unix%> make install
unix%> ldconfig
unix%> make clean
```

Note that here we have built S-lang using the GNU version of readline (part of the required packages listed above).

6) We now download ISIS from http://space.mit.edu/ASC/ISIS/download.html. The current version is 1.4.9-55. As with the other packages, we unpack it to:

```
/usr/local/isis-1.4.9-55/
```

In building ISIS, for simplicity we have it use the cfitsio (data reading) and pgplot (data plotting) packages built with the HEASOFT installation we just created. Furthermore, we set the default spectral models to come from the HEASOFT XSPEC 12 package (as opposed to the HEASOFT XSPEC 11.3.2 package; there are plans to deprecate XSPEC 11 in future HEASOFT releases).

```
unix%> cd /usr/local/isis-1.4.9-55/
unix%> setenv HEADAS /usr/local/heasoft-6.6.2/i686-pc-linux-gnu-libc2.9/
unix%> ./configure --with-headas=$HEADAS --with-xspec-version=12
unix%> make
unix%> make install
unix%> make check
unix%> make clean
```

Note that in the above we have run a test ("make check") of the ISIS installation. This runs a suite of programs in the " /usr/local/isis-1.4.9-55/test " directory that exercise a variety of different data analysis functions. If all of the above has worked well, then there should be no failures in the tests.