

Mac OS X 10.5 & 10.6 Installation Instructions

[Credits: Mike Nowak, Peter Polko, Sera Markoff, Salomé Dibi, Montserrat Armas Padilla]

Throughout this writeup "\$" signifies a terminal prompt!

%%
%%

X-ray install stuff

%%
%%

The following is a set of instructions for building HEASOFT 6.9,
gnu redline 6.1,

S-lang 2.2.2, and ISIS 1.6.0-3 and slirp 1.9.8 on a Apple OS X
10.5 or 10.6

system. Building on Apple systems is *definitely* trickier than

building on a Linux system. Isn't Apple OS X just the same as
Linux?

No, it's not! It's 95% the same, but 5% of the time it's
something

subtly different that causes all sorts of problems.

For that reason, we ask that you follow the instructions **exactly** as

you see them here. Please, no improvising! If everyone starts from

the same point, this will minimize the number of variations on

problems that we have to track down and fix.

All the steps below should take about two hours. If it's taking

substantially longer than that, there's probably some problems,
so

contact us!

First line of help for Xray related software *only* = Mike Nowak
(mnowak@space.mit.edu), please CC s.b.markoff@uva.nl as
well.

Usually one would not download and build everything into `/usr/local/src`

but we have found that doing it this way seems to avoid some of the library/communication problems. So please stick to this even if it seems weird to you!

1) Download all the necessary source code into `/usr/local/src` (create the "src" directory as root if it does not exist):

```
$ cd /usr/local
```

```
$ sudo mkdir src
```

--HEAsoft 6.9 (<http://heasarc.gsfc.nasa.gov/lheasoft/download.html>)

In step 1 select "Source code distribution". Then select the right platform (for example "Mac Intel - Darwin 9.x (OS X 10.5.x)" but check the "Hardware" tab in your System Profile if you are unsure).

In step 2 select "All" and click the "Submit button". The full download is about 700 MB.

--gcc & gfortran binary tarballs from <http://hpc.sourceforge.net/>

(this is new: ☐ only the Leopard compilers seem to work on

Snow Leopard to correctly compile Heasoft6.9!!)

Leopard PPC: <http://prdownloads.sourceforge.net/hpc/gcc-leopard-ppc-bin.tar.gz?download>

Leopard Intel: <http://prdownloads.sourceforge.net/hpc/gcc-leopard-intel-bin.tar.gz?download>

**note: even if you have gcc/gfortran from another source, you need the above versions!!

--GNU readline 6.1 from <http://tiswww.case.edu/php/chet/readline/rltop.html>

Direct link: <ftp://ftp.cwru.edu/pub/bash/readline-6.1.tar.gz>

--S-Lang 2.2.2 from <http://www.s-lang.org/download.html>

Direct link: <ftp://ftp.fu-berlin.de/pub/unix/misc/slang/v2.2/slang-2.2.2.tar.gz>

-ISIS 1.6.0-12, download isis-1.6.0-12.tar.gz from: <http://www.mit.edu/~houck/isis/>

--SLIRP 1.9.8 from <http://space.mit.edu/cxc/software/slang/modules/slirp/>

Direct link: <ftp://space.mit.edu/pub/mnoble/slirp/slirp-1.9.8.tar.gz>

--S-Lang statistic module from <http://space.mit.edu/cxc/software/slang/modules/stats/>

Direct link: <http://space.mit.edu/cxc/software/slang/modules/stats/src/slstats-1.0.2.tar.gz>

2) Upgrade to the newest Xcode version

(if Xcode is not already installed on your computer, install Xcode off the Mac OS X disk you got with your laptop)

Then run Software Update (possibly repeatedly) until all software is up to date. On your Mac, this is done by going to the "Apple" sign at the upper left of your top menu bar and selecting Software Update. Once you do this once, often requiring restart, you should try again until the program tells you there are no further updates to install.

3) Using compatible, recent and stable compilers is crucial. For Heasoft6.8 I had used Macports to install the gcc/gfortran compilers, so if you have done the same, the first thing you need to do is:

```
$ sudo port uninstall gcc44
```

And just to be safe I moved /opt to /opt_hide.

4) Start a new session as root to do the important stuff, make sure once you do this command you perform all the following tasks in this window

```
$ sudo tcsh
```

(for cshell...most of these instructions are assuming cshell but can be adapted for bash).

In some cases if you have problems you may need to do:

```
$ sudo -E tcsh
```

or

```
$ sudo -E bash
```

For all of the following steps you must be working in this terminal, with root access. You can always check who you are by typing

```
$ whoami
```

and the answer should be "root" if all is well.

4.5) Remove any existing builds of these programs below that you have (in whatever directories) to avoid conflicts

5) Install gcc and gfortran in the top directory, it will unpack everything properly into the /usr subdirectories. The binaries themselves will be in
/usr/local/bin

```
$ cd /
```

```
$ mv /gcc-leopard-intel-bin.tar.gz .
```

```
$ gunzip gcc....tar.gz
```

```
$ tar xvf gcc....tar
```

6) The most important step for *all* following processes is to make sure that all installations are performed using the new compilers. This means that you need to make 100% sure that the \$PATH variable, which tells your computer where to look for things, is set up correctly.

type:

```
$ echo $PATH
```

Normally this will look like:

".:...:/bin:/sbin:" etc, with a long list of directories. What you want to do is make sure that `/usr/local/bin` appears in that list after `/bin` and

/sbin

but before everything else like

/sw/bin

or

/opt/local/bin

, which are the top level directories that programs like Fink and Macports set up to put software in. The point is that those programs may have given you versions of gcc/gfortran, but it's important that you get select from

/usr/local/bin

first!!

So you may need to hand-edit .cshrc or .bashrc so the path looks like:

".:...:/bin:/sbin:/usr/local/bin:"....the rest of the path after this

Keep in mind that your \$PATH may not look like the next person's \$PATH, this is something that is customizable and depends also what programs you have used to install things, if at all. So no matter what is in there, just make sure /usr/local/bin is one of the first places it looks!

7) Now that the path is correct, edit your .cshrc to set the compiler environments that Heasoft will use:

```
setenv FC "/usr/local/bin/gfortran"
```

```
setenv CC "/usr/local/bin/gcc"
```

```
setenv CXX "/usr/local/bin/g++"
```

```
setenv GCC "/usr/local/bin/gcc"
```

(or equivalent in .bashrc):

```
CC="/usr/local/bin/gcc"; export CC
```

```
FC="/usr/local/bin/gfortran"; export FC
```

```
CXX="/usr/local/bin/g++"; export CXX
```

```
GCC="/usr/local/bin/gcc"; export GCC
```

If you did not do the preceding steps carefully, everything below will fail!!!

Now source the files:

```
$ cd
```

```
$ source .cshrc (or .bashrc)
```

to make sure the changes are in place before the next steps

8) unpack heasoft:

```
$ cd /usr/local/src
```

```
$ tar xvf heasoft6.9src.tar
```

9) The next step is very important, you need to set the permissions within the build directories to be writeable, which for some reason does not happen automatically with this version.

```
$ cd heasoft6.9
```

```
$ chmod -R a+w BUILD_DIR
```

10) now set up the configuration, making a log file to send us if it doesn't work

```
$ cd BUILD_DIR
```

```
$ ./configure --x-libraries=/usr/X11/lib  
--x-includes=/usr/X11/include >& config.out &
```

[If your X11 libraries are somewhere else, fill in the appropriate path]

11) Now you can start to build, making log files that you will send us if all goes amok.

```
$ make >& build.log &
```

12) And install


```
$ make install >& install.log &
```

```
$ make clean
```

(good way to clean up files after installation)

13) edit `.cshrc` (or do equiv in `.bashrc`) to have the line:

```
setenv HEADAS  
/usr/local/src/heasoft-6.9/i386-apple-darwin10.3.1
```

Or equivalent for your architecture. You will find the correct path by looking in /usr/local/src/heasoft-6.9 for the directory with a similar name as above

This sets the paths you will need to make Isis "see" Heasoft

14) unpack readline-6.1.tar in /usr/local/src, roughly following the same steps as above

```
$ cd /usr/local/src
```

```
$ tar xvf readline-6.1.tar
```

```
$ cd readline-6.1/
```

```
$/configure >& config.out &
```

```
$ make >& build.log &
```

```
$ make install >& install.log &
```

```
$ make clean
```

15) install slang

```
$ cd /usr/local/src
```

```
$ tar xzvf slang-2.2.2.tar.gz
```

```
$ cd slang-2.2.2/
```

```
$ ./configure --with-readline=gnu >& config.out &
```

```
$ make >& build.log &
```

```
$ make runtests
```

```
$ make install >& install.log &
```

```
$ make clean
```

16) install slirp-1.9.8

```
$ cd /usr/local/src
```

```
$ tar xzvf slirp-1.9.8.tar.gz
```

```
$ cd slirp-1.9.8/
```

```
$ ./configure >& config.out &
```

```
$ make >& build.log &
```

```
$ make install >& install.log &
```

```
$ make clean
```

17) cross your fingers, because if this works, you're almost done! The last (and often trickiest) step is installing isis

```
$ cd /usr/local/src
```

```
$ tar xzvf isis-1.6.0-12.tar.gz
```

```
$ cd isis-1.6.0-12
```

```
$ ./configure --with-headas=$HEADAS --x-libraries=/usr/X11/lib
```

```
--x-includes=/usr/X11/include >& config.out &
```

```
$ make >& build.log &
```

```
$ make install >& install.log &
```

```
$ make clean
```

18) last little module for statistics

```
$ cd /usr/local/src
```

```
$ tar xzvf slstats-1.0.2.tar.gz
```

```
$ cd slstats-1.0.2
```

```
$ ./configure >& config.out &
```

```
$ make >& build.log &
```

```
$ make install >& install.log &
```

```
$ make clean
```


%%
%%

Radio: AIPS

%%
%%

**note, as compared to the X-ray instructions above, you want to run all this as yourself, not as root!! So open a new window and make sure when you do \$whoami it does not say "root"

1) Download install.pl from <http://www.aips.nrao.edu/dec10.shtml>,

(Direct link: <ftp://ftp.aoc.nrao.edu/pub/software/aips/31DEC10/install.pl>)

where you can find detailed instructions beyond this.

2) Decide where you want to install aips. I would suggest setting up a directory \$HOME/aips. Put install.pl in this directory and run

```
$ perl install.pl -n
```

Read & follow the on-screen instructions. Refer to the more detailed notes [here](#) at each step. The following options seem to work for me, but you may wish to customize them. Note that the numbers are not actually consecutive(!)

Screen 0 -- hit enter

Screen 4 -- hit enter (ie accept the default choice)

Screen 4b -- hit enter (ie, allow yourself to have write access)

Screen 5 -- choose a name; can be whatever you like

Screen 5a -- enter "y"

Screen 5b -- accept the default (MACINT for intel chip)

Screen 6 -- hit enter to accept LOCALHOST

Screen 7 -- add the data area suggested, make sure it matches where your data area is defined if not default

Screen 8 -- press "A" to accept current settings, then type A4

Screen 9 -- press "A" to accept current settings

Screen 9b -- press "A" to accept current settings

Screen 11 -- press "A" to accept, then "Y" to start the installation.

You will now need to wait while the data files are downloaded and installed. (can take a long time depending on your connection speed).

Eventually, the configuration system will set up configuration for a "midnight job" (MNJ; ie, an automatic update of the AIPS files): just hit return to acknowledge the default settings, then again to enter a blank CVS password (as instructed), then again when "ready to proceed" (you can examine the do_daily.localhost

file if you plan to run a midnight job). You will then be prompted to hit enter for a fourth time(!) before finally being told:

AipsWiz: That's it. You should now have the latest AIPS!
Enjoy.

Edit your `$HOME/.bashrc` or `$HOME/.cshrc` to make an alias to initialize AIPS when you want to use it by adding the line:

```
alias useaips "source $HOME/aips/LOGIN.CSH" (for cshrc,  
LOGIN.SH for bash)
```

(or fill in correct path to `LOGIN.CSH` or `LOGIN.SH` for `bashrc`, as appropriate for your install).

Modify your `/etc/services` file (as root, ie., for this part you need to go back to the root terminal or set up a new one) to have the following lines:

```
sssin      5000/tcp      SSSIN     # AIPS TV server
```

ssslock 5002/tcp SSSLOCK # AIPS TV Lock

msgserv 5008/tcp MSGSERV # AIPS Message
Server

tekserv 5009/tcp TEKSERV # AIPS TekServer

aipsmt0 5010/tcp AIPSMT0 # AIPS remote FITS
disk access

aipsmt1 5011/tcp AIPSMT1 # AIPS remote tape 1

aipsmt2 5012/tcp AIPSMT2 # AIPS remote tape 2

aipsmt3 5013/tcp AIPSMT3

aipsmt4 5014/tcp AIPSMT4

aipsmt5 5015/tcp AIPSMT5

aipsmt6 5016/tcp AIPSMT6

aipsmt7 5017/tcp AIPSMT7

Now you are ready to run AIPS. The cookbook can be viewed online [here](#) and as a pdf [here](#)

%%
%%

Radio: CASA

%%
%%

download the appropriate disk image and install by putting CASA in your applications folder. Run once (make sure your account has administrative abilities) to allow CASA to do some initial setup.

https://svn.cv.nrao.edu/casa/osx_distro/10.5/CASA-intel-11761.dmg

https://svn.cv.nrao.edu/casa/osx_distro/10.6/CASA-intel-11761.dmg

%%
%%

Fermi install stuff

%%
%%

Download the appropriate MacOS10.5 or 10.6 binary from:

<http://fermi.gsfc.nasa.gov/ssc/data/analysis/software/>

follow install directions on

http://fermi.gsfc.nasa.gov/ssc/data/analysis/software/binary_install.html

%%%%%%%%
%%%%%%%%

And one more....DS9 for imaging data:

All platform binaries downloadable from: <http://hea-www.harvard.edu/RD/ds9/>

Direct downloads for the X11 versions:

for OS version 10.6 (Snow Leopard): <http://hea-www.harvard.edu/saord/download/ds9/darwinsnowleopard/ds9.darwinsnowleopard.6.1.tar.gz>

for OS version 10.5 (Leopard): <http://hea-www.harvard.edu/saord/download/ds9/darwinleopard/ds9.darwinleopard.6.1.tar.gz>

Untar the archives to find two files: a binary executable ds9 and a ZIP file ds9.zip. Keep these two files together!

To make sure this is on the path for Heasoft to find, take one more step: you need to put both files in a directory already in your path (like /usr/local/bin), or add the directory wherever

those two files are to your path. For instance:

```
$ cd ds9.darwinsnowleopard.6.1  
$ sudo cp ds9 ds9.zip /usr/local/bin
```

or add to your .bashrc:

```
PATH="$PATH:/Users//ds9.darwinsnowleopard.6.1"; export  
PATH
```

or .cshrc:

```
setenv  PATH  "$PATH:/Users//ds9.darwinsnowleopard.6.1"
```