Antelope on Mac OS X

Trials and Tribulations

What's the big deal?

- Mac OS X is just Unix, right?

Goals

- Initially, we want to make it possible to work with Antelope on Macintosh laptops
- Not aimed at running networks (Sun workstations)
- or as a field processor (New marmot)
- not trying for Cocoa interface, just compatibility among platforms
- same source tree for Solaris, Linux/X86, Linux/Arm, Darwin

Problems:

- Mac OS X is based on mixture of BSD Unix, Mach and NeXt OS.
- Apple is aiming toward least sophisticated users
  - sometimes good, sometimes not so good
- Case *insensitive* file system
- filesystem mounts are different; paths are different

Link problems

- standard libraries are different: everything is thrown together into libc
- linking is much more complicated: two kinds of dynamic libraries, lots of options
- ldd missing
- dlopen() missing

Implementation problems:

- no /proc filesystem: much more difficult to get system performance statistics
  - hard to get exec line for error routines
  - hard to keep track of children pids (eg, w/rtexec)
  - other info about pid – %cpu, etcetera - extremely hard to get

More differences:

- tell missing
- poll() missing
- statvfs() different
- cuserid vs getlogin
- ifconfig is different

Even time is different

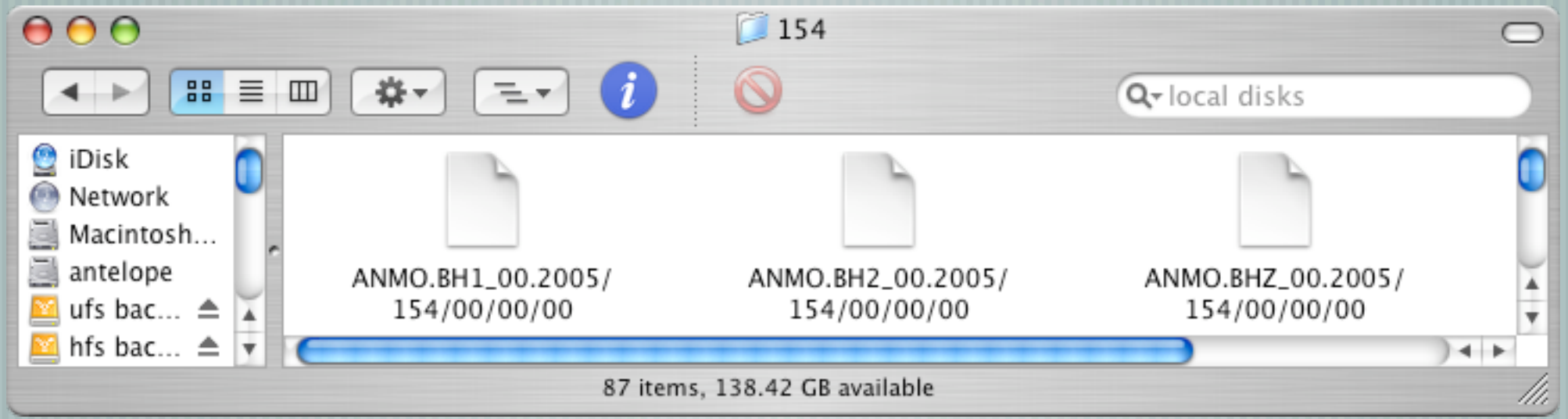
- time() different
- timers are different
- timezone methods different

development problems:

- X11 available separately
- Development environment available separately
- Some Unix tools not available: eg, TclX
- Fortran available separately
- rm -f returns error code

Unresolved Issues

- Individual process statistics (eg, rtm)
- matlab: linking issues, namespaces and other complications
- waveform file names: finder doesn't like colons



Getting Started

- Download X11 from Apple and install
- edit /etc/ssh*config
  - X11Forwarding yes
  - ForwardX11 yes

Possibly useful hints

- edit /etc/sysctl
  - kern.maxproc=2048
  - kern.maxprocperuid=1024
  - `sudo sysctl -w kern.maxproc=2048`
  - `sudo sysctl -w kern.maxprocperuid=1024`
- look at *hdiutil* to create a ufs volume (which understands the difference between upper and lower case)

Compiling your own?

- Download and install developer toolkit Xcode from Apple
  - Remember to install X11SDK, using *Customize* button

Fortran? (oh horrors)

- Install g77 from hpc.sourceforge.net
- hide dynamic libraries libg2c*.dylib
  - so linker uses static libs, and programs are portable to machines w/o g77

Non-Antelope items

- Use Netinfo Manager to change shell to tcsh
- put terminal and x11 in dock
- block pop up windows in Safari
- Download, install and configure MenuMeters
  - http://www.ragingmenace.com/software/menumeters
- Set up your backups

Install Antelope

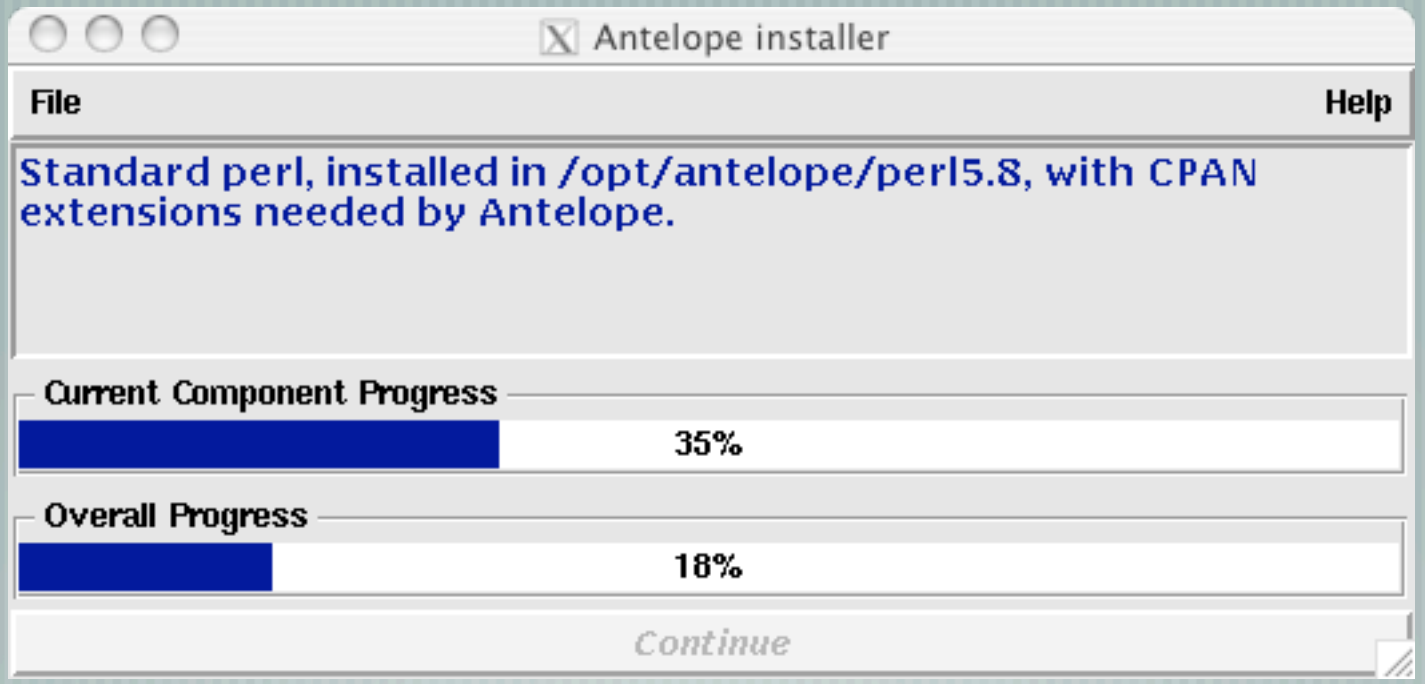
- Insert Antelope cd, and run X11
- cd /Volumes/Antelope_4.7
  - ./Install_antelope
  - without X11, Install_antelope -ut

```
shasta% cd /Volumes/Antelope_4.7
shasta% ls
Changes                 Linux/                  _4.7.2005_Apr_14_12:23
Darwin/                 README                  common/
Install_antelope*       SunOS/                  license.txt
```

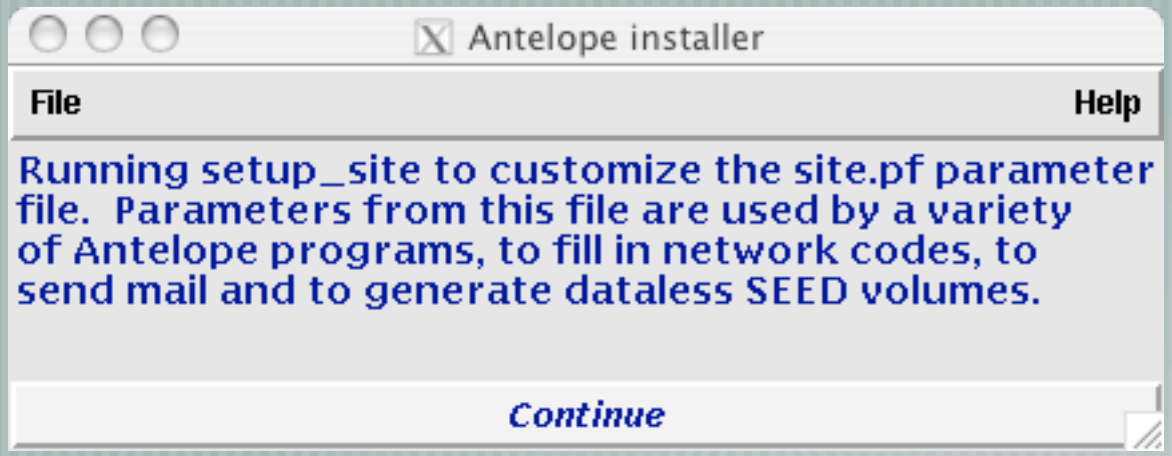
Installing Antelope

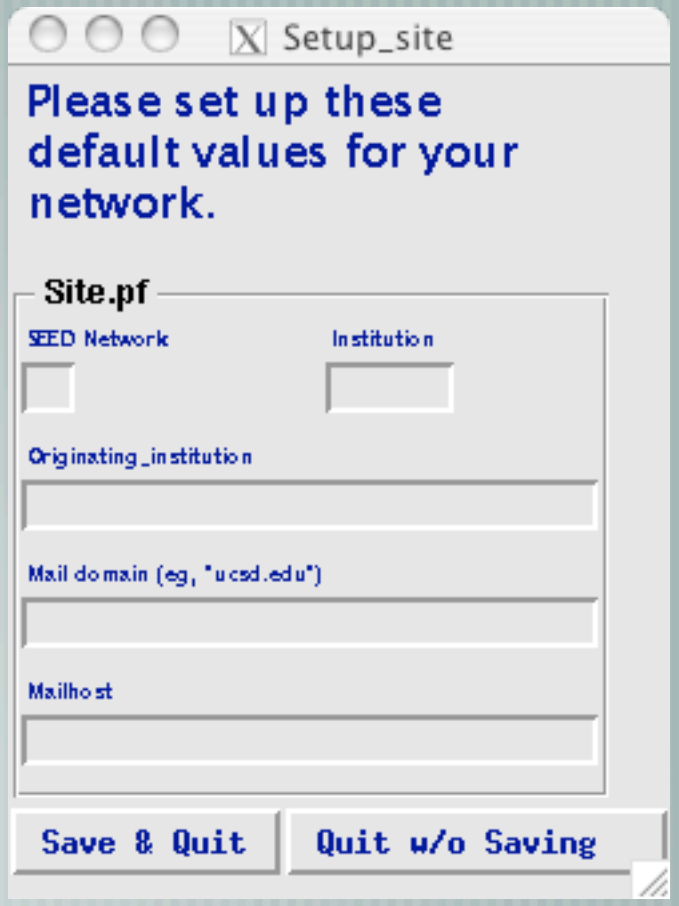
Get licensed

Watch the install

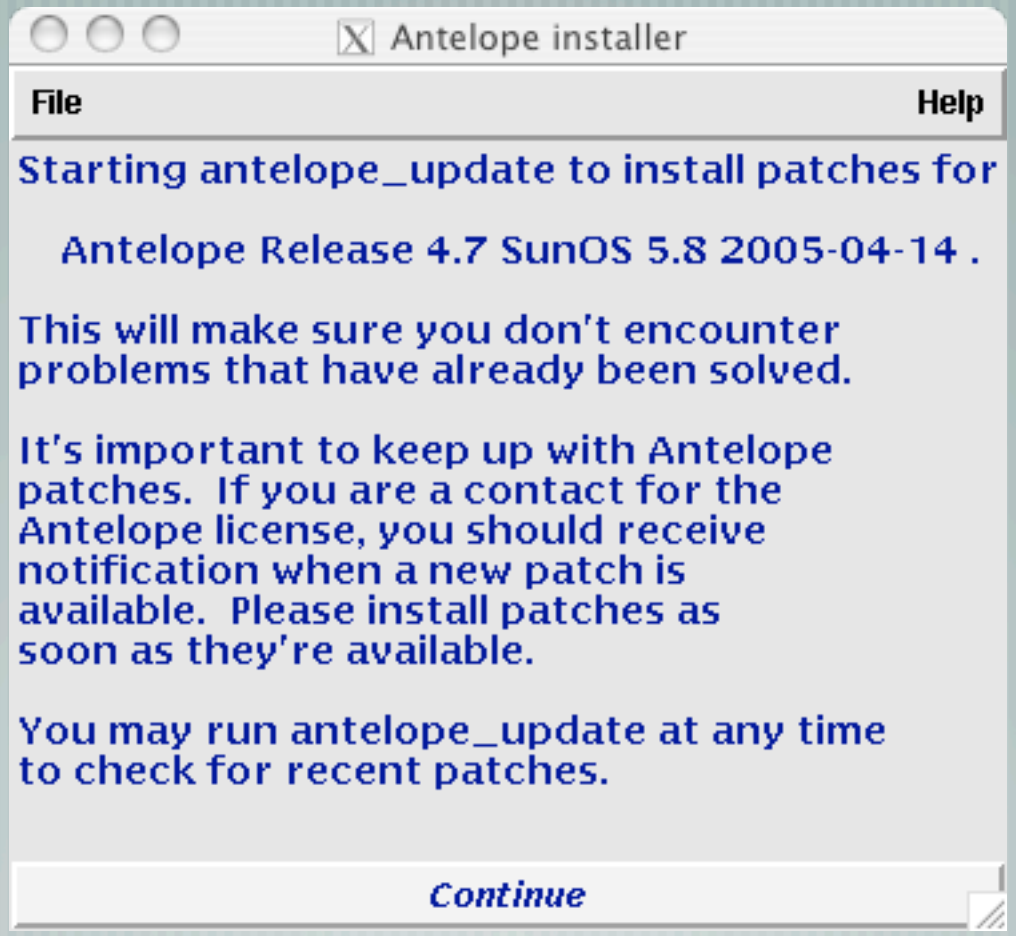


Customize

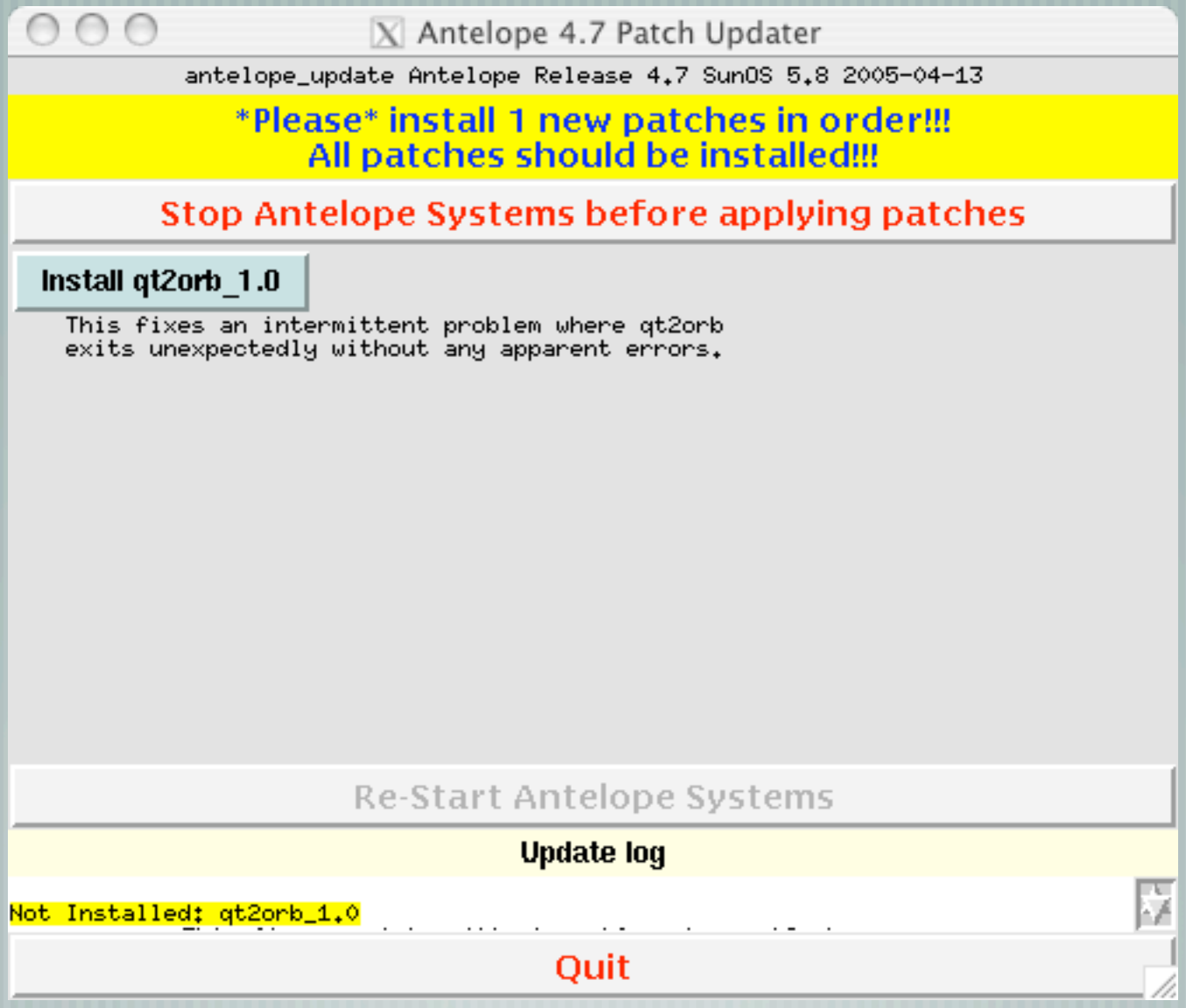

Antelope installer
File
Help
Running setup_site to customize the site.pf parameter file. Parameters from this file are used by a variety of Antelope programs, to fill in network codes, to send mail and to generate dataless SEED volumes.
Continue


Setup_site
Please set up these default values for your network.
Site.pf
SEED Network
Institution
Originating_institution
Mail domain (eg, "ucsd.edu")
Mailhost
Save & Quit
Quit w/o Saving

Apply patches



antelope_update



Still need to install license

What about Tiger?

- Sometimes observed: wc -l < /dev/null
  - segmentation violation
- g95 compiler is pickier about fortran
- sigignore() function added

My guess is that this an instance of the new behavior ("not a bug", according to Apple) of dyld on Tiger that makes binaries crash that were built and linked correctly: The new dyld has the idea of trying to resolve all undefined symbols in all dependent libraries, whether the symbols are needed for the executable or not.

Until MacOSX 10.3, you could be reasonably sure that if an executable did not produce an "undefined symbols" error at link time, it would not produce one at run time, either - ld and dyld had almost identical strategies for resolving symbols. On Tiger, this is no longer the case - you can very easily produce executables that link OK but crash at runtime.

Problem:

```
X Error of failed request:  BadAtom (invalid Atom parameter)
  Major opcode of failed request:  20 (X_GetProperty)
  Atom id in failed request:  0xb9
  Serial number of failed request:  12
  Current serial number in output stream:  12
```

Solution:

```
alias ssh ssh -Y
```

Probably a new version of ssh problem.

Huge changes in Antelope

- references in parameter files
  - eg,

```
dir     data/&sta/&chan

sta     AKA

chan    BHE
```

Huge changes in Antelope

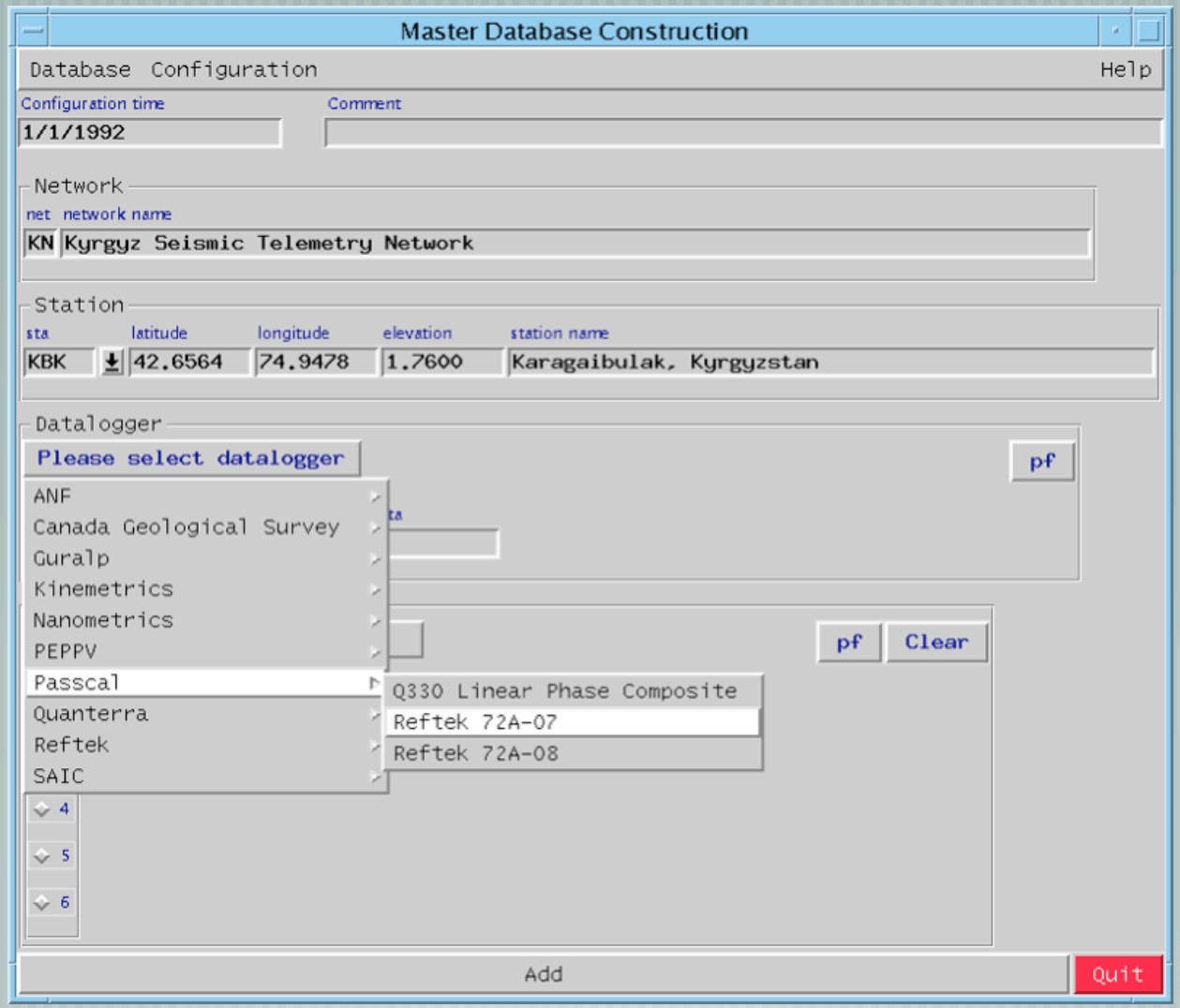
- schema table extensions
  - additional tables are described in css3.0.ext directory
  - rt1.0 is no longer needed
  - dbverify, dbe parameter files simplified

Huge changes to come

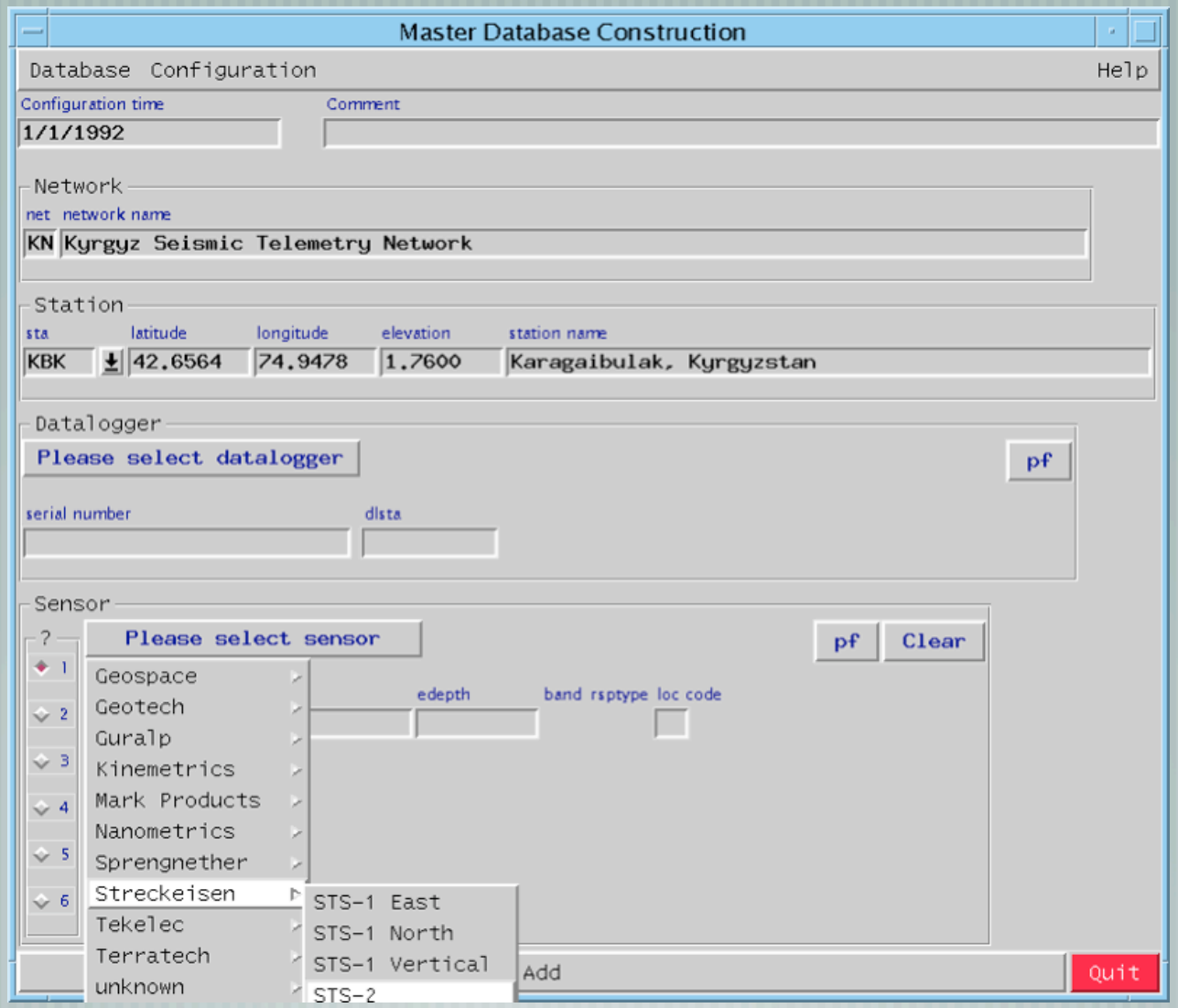
dbbuild

- new menu organization for dataloggers and sensors
- individual sensitivities by sensor output
- individual gains by data logger input leads
- serial number lookup of sensitivities and gains

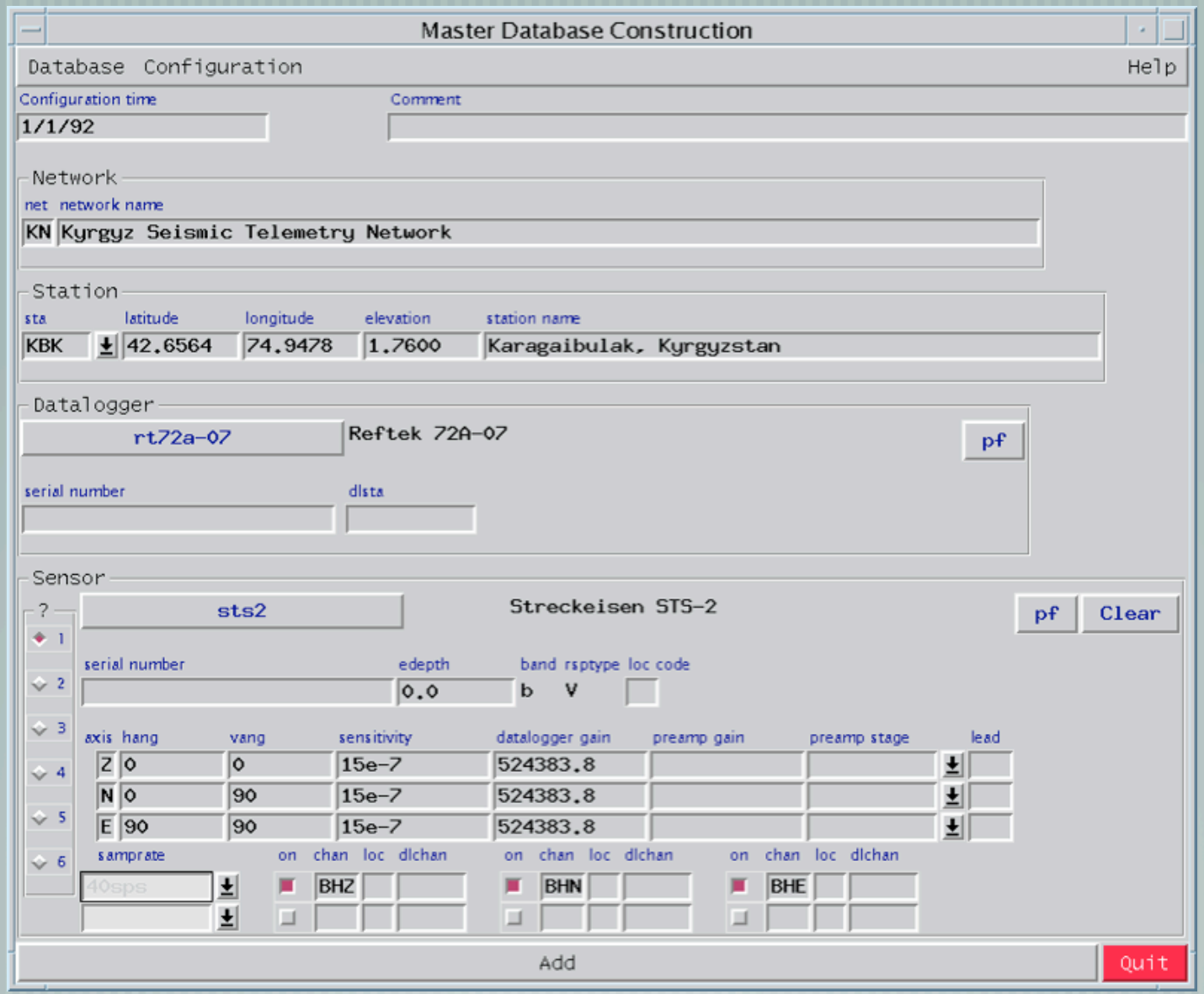
New menu for dataloggers



New menu for sensors



Sensitivities and gains



new lag statistic

```
 lag    thread   %pkt   pktid  who@where                    command
 0.00    99921  0.489  2363162 rt@          orblatency -m AZ.* -p 300 :anza db/anza
 0.00    99618  0.489  2363162 rt@          orb2db -v -m match -r reject -S state/orb2dbAZ :anza
 0.00       15  0.489  2363161 rt@mw        orbmonrtd
 0.00       20  0.489  2363161 rt@mw        orbmonrtd
 0.00       67  0.489  2363161 rt@          orbdetect -out :anza :anza db/anza
 0.00       23  0.489  2363161 rt@mw        orbmonrtd
 0.00       21  0.489  2363161 rt@bbarray   orb2orb 132.239.4.29:anza : -m AZ.* -r reject
 0.00       84  0.489  2363161 rt@          par2db -i 60 :anza param/anza_par
 0.00   154507  0.489  2363161 rt@igpprt    orbmonrtd
 0.00       18  0.489  2363133 rt@csulb.edu orb2orb -S state/orb2orb_AZdb -m match -r .*/MGENC/.*|.*/log
 0.00   158041  0.489  2363118 danq@agate   orbstat -i igpprt: lag
 0.00       19  0.489  2363116 rt@mercali   orb2orb igpprt:anza :status -m match
 0.00    99624  0.489  2363110 rt@          orb2db -m match -S state/orb2dbCI :anza db/anza
 0.00       14  0.489  2363106 rt@bbarray   orb2orb igpprt:anza :demo -m AZ_BV.*
 0.00   157911  0.489  2362085 rt@          orbassoc -start OLDEST -v -select /db/detection :anza :anza ttgrid-
local_regional ttgrid-tele
 0.00   157914  0.489  2362085 rt@          orb2dbt -state state/orb2dbt -overwrite :anza db/anza
 0.00   157969  0.488  2360244 rt@mercali   /opt/antelope/4.7/bin/orbpf2db igpprt:anza db/anza_orbregistry
 0.01       17  0.486  2346353 rt@csulb.edu orbxfer -v -d db 132.239.4.29:anza
 0.07   154504  0.459  2216489 rt@          /opt/antelope/4.7/bin/awish /opt/antelope/4.7/bin/rtm --
 0.44   157917  0.298  1442123 rt@lindquist orb2dbt -reject /db/(detection|trigger) -state state/orb2dbt
-overwrite igpprt db/anza :
 0.54   158039  0.257  1240299 rt@          orbmag -state state/orbmag :anza :anza db/anza
 1.00       24 -0.000       -1 rt@          qt2orb -dataorb :anza -cmdorb :anza -calib_db db/anza
 1.00       25 -0.000       -1 rt@          qt2orb -dataorb :anza -cmdorb :anza -calib_db db/anza
max lag =  1.00
```