Kent Lindquist White Paper

Number 2002-001

AEIC Antelope Upgrade from 4.4u-β to 4.4

Kent Lindquist

February 8, 2002

Antelope Upgrade from 4.4u Beta to 4.4 Production Release

The following report tracks the steps I am taking to upgrade Antelope from the Beta Release of 4.4u, which we have been running since December 2001, to the Antelope 4.4 production release dated January 24, 2002. Note that with this version of Antelope, the distinction between the University edition and the full edition has been abolished.

First, we will insert the SPARC production release CD on nordic. After a *volcheck* (usually not necessary), I get the file manager for the CD, and click on the icon for "*start*" (an executable script on the CD). Changes between the last release and the current one are described in the *Changes* file on the CD. This start script should launch a self-explanatory TCL/TK Graphical User Interface. Proceed with installation. There was one snag, in that the previous compile of the ImageMagick/PerlMagick tools into the 4.4u perl were done as root, thus we have to fix the ownership for the file copying to work:

```
nordic# chown -R kent:bdogs /opt/antelope/perl/
nordic#
```

After this I went in and installed the perl, data, and doc directories by hand (rerunning the *start* script).

Before proceeding with the licensing stage, I took the license email that Danny Harvey already sent me and saved it, creating a proper license file from it:

```
nordic% cat license.pf
license_key     &Tbl{
  Nda9bd02113cb019a2d27d7db8975c9ca 6/30/2002 # Antelope 4.4 network 137.229.32.0
  N6ec94d0ab33c88a52054eecf930e559a 6/30/2002 # Antelope 4.4 network 137.229.32.0
}
nordic%
```

Danny Harvey of BRTT, Inc. has also sent node-locked licenses for the machines beam (in Anchorage), plate (laptop), alaskavsn (in Canada at the Pacific Geoscience Center), and alaskaseis (in Hokkaido, Japan). Note that the Antelope 4.4 software will not work with anything earlier than Solaris 2.7.

The next step is to install the newly announced patch cluster for Antelope 4.4:

http://www.brtt.com/patches/4.4

I downloaded this to

/home/kent/work/antelope/patches/4.4/assortment1.0.tar.gz

and did

```
gunzip assortment1.0.tar.gz
tar xvf assortment1.0.tar
```

Next, we need to apply our local modifications to Perl to the new Antelope Perl. For further discussion of this, see "AEIC Antelope Upgrade from 4.3u to 4.4u," *Kent Lindquist White Paper 2001-004*. It's necessary to do this in short order since the *dbrecenteqs* software relies on this set of objects.

```
nordic# cd /opt/free/src/IMAGEMAGICK/ImageMagick-5.3.0
nordic# cat CONFIGURE_COMMAND
./configure --enable-lzw --enable-shared --prefix=/opt/free --with-perl=/opt/antelope/perl/bin/
perl CC=cc CFLAGS="-I/opt/sfw/include -I/opt/free/include -xCC" LDFLAGS="-R/opt/sfw/
lib -L/opt/sfw/lib -R/opt/free/lib -L/opt/free/lib" CPPFLAGS="-I/opt/sfw/include -I/opt/free/
include -xCC" --without-magick-plus-plus
nordic# source CONFIGURE_COMMAND
configuring ImageMagick 5.3.0
.....
nordic# make
Making all in delegates
Making all in coders
Making all in magick
Making all in utilities
Making all in tests
nordic# make install
Making install in delegates
/bin/sh ../mkinstalldirs /opt/free/lib/ImageMagick
....
```

Finally, to avoid the mistake of last time, change ownership of the installed package:

```
nordic# chown -R kent:bdogs /opt/antelope/perl/
nordic#
```

Next we will update the environment variables in order to support the new release. For my personal setup files, this is as simple as changing

```
setenv ANTELOPE /opt/antelope/4.4u
```

to

```
setenv ANTELOPE /opt/antelope/4.4
```

in /home/kent/setup/envrc. It is important to know about this since some of the automatic-system accounts (ppicker in multiple home directories; uafarr; bbanddat) use a similar setup.

All the rest of the variables (ANTELOPEMAKE, AEIC, AEICMAKE, PATH, MANPATH, SCHEMA_DIR, FOREIGNKEYS_DATABASE, DBMAPEVENTS_DATA, PFPATH,

GRX_FONTPATH) are set to rely on this one. Note the relatively new pair of variables AEIC and AEICMAKE. These point to the local distribution of AEIC software (containing everything that is engineered to integrate properly with Antelope). These depend on the Antelope version since this software tree is compiled against Antelope.

Next we handle the Antelope contributed code. Binaries and compiled libraries are already distributed with Antelope, therefore they are all installed. However, we like to keep the contributed source-code distribution available locally for further development and debugging. The proper way to update this would be:

```
nordic% cd /opt/antelope/4.4
nordic% mkdir contrib
nordic% cd contrib
nordic% setenv CVSROOT aesn.geology.indiana.edu:/opt/antelope/cvs
nordic% setenv CVS_RSH ssh
nordic% cvs checkout -d src contrib
nordic% make Include
nordic% make install
```

Note that at AEIC, CVS_RSH is set correctly for Kent. However, it should also be set for all other CVS operations at AEIC. This presumes the user doing the CVS has an account on the Indiana machine aesn.geology.indiana.edu (point of contact is Dr. Gary Pavlis, pavlis@indiana.edu).

Unfortunately, it is not at this point possible to perform this CVS operation, since the aesn machine is down due to break-in. Even though the following step is a very bad approach, I know we will get away with it this time. Thus:

```
nordic% cd /opt/antelope/4.4
nordic% cp -r /opt/antelope/4.4u/contrib .
```

Once again, I do not recommend this. For the sake of completeness I will recompile:

```
nordic% cd /opt/antelope/4.4/contrib/src/
nordic% make Include
```

This uncovered a subtle problem with several of the Makefiles for the Matlab Antelope toolbox. I fixed that, and now these changes will have to be submitted to Indiana as soon as their server comes back up.

```
nordic% make install
```

This make includes the Antelope Toolbox for Matlab. In order for the Matlab Antelope Toolbox to work with the new version, we need to run:

```
nordic# install_matlab_antelope_links
nordic#
```

There's a man page on this tool if needed.

The next step is to update the AEIC software distribution. First, we want to close out any outstanding current changes:

```
nordic% cd /usr/local/aeic/4.4u/src/
nordic% cvs commit
```

I had to make one fix in order to get this to work. There's a copy of aeic_rtsys_blueprint.fm in the src/adm/aeic_rtsys_blueprint directory, but it's not as current as the distribution (probably because Trilby is working on it somewhere else). Therefore I did

```
nordic% cd adm/aeic_rtsys_blueprint/
nordic% cvs update -d -P .
```

and I can repeat

```
nordic% cd /usr/local/aeic/4.4u/src/
nordic% cvs commit
```

with no complaints. Now, to label the end of development that was done in strict compatibility with Antelope 4.4u, we tag the whole distribution. This step is not required, however it makes rollback much more convenient if, after future development, we have to checkout an earlier version from CVS that will be compatible with 4.4u.

```
nordic% cd adm/aeic_rtsys_blueprint/
nordic% cvs rtag last_44u src
```

Now we start moving forward again, creating a new checkout of the AEIC source that will work with and expand upon Antelope 4.4:

```
nordic% cd /usr/local/aeic
nordic% mkdir 4.4
nordic% cd 4.4
nordic% setenv CVSROOT /usr/local/aeic/cvs
nordic% setenv CVS_RSH ssh
nordic% cvs checkout src
```

Once again, note that I usually keep CVS_RSH set in my setup files so I don't have to do this explicitly. Also, note that the root of the CVS distribution is in /usr/local/aeic/cvs, with no "4.4" or "4.4u" etc. This is because the repository itself is separate from individual versions that are checked out. For convenience, we keep the real repository in a directory parallel to the checked-out distributions that work with each release of Antelope. There's one thing I have to do by hand before continuing: we need to install the correct makerules before trying to compile the new distribution.

```
nordic% cd /usr/local/aeic/4.4
nordic% mkdir include
nordic% cd include
nordic% cp /usr/local/aeic/4.4u/include/aeicmake .
```

Inside /usr/local/aeic/4.4/include/aeicmake we change

```
DEST=/usr/local/aeic/4.4u
```

to

```
DEST=/usr/local/aeic/4.4
```

There may be other more automatic options for handling this, perhaps relying on a (correctly set!!) AEIC environment variable. However, for now this one manual step is a reasonable stepping stone. The rest--directory creation etc.--will now be automatic:

```
nordic% cd /usr/local/aeic/4.4/src/
nordic% make Include
```

Before running this, I double checked:

```
nordic% echo $AEIC
/usr/local/aeic/4.4
nordic% echo $ANTELOPE
/opt/antelope/4.4
nordic% echo $AEICMAKE
/usr/local/aeic/4.4/include/aeicmake
nordic% echo $ANTELOPEMAKE
/opt/antelope/4.4/include/antelopemake
nordic%
```

There are several problems due to programs that I only incompletely removed when I moved them to the Antelope contrib cvs. I had to fix several subdirectories with

```
nordic% cvs update -d -P .
```

Now we can do:

```
nordic% make Include
nordic% make install
```

This uncovered a couple problems. First, *aeic_dbap* needed its call to *zopen* updated. Also, *banner* is only available for Antelope programs, so I commented it out. I installed *fixman* with

```
nordic% cd /opt/antelope/4.4
nordic% tar xvf /home/kent/work/antelope/patches/4.3u/fixman.tar
```

then used it to fix a variety of unhealthy man pages in /usr/local/aeic/4.4. I also changed *banner* to *cbanner* in win2db.c. I tweaked the makefile in bin/rt/orb2vdl/vdl to force recompilation of *makedate*, which failed because it was linked against the Antelope 4.3u libraries. In fact, we can do the entire distribution with the *find* combined with the *fixman* commands

```
nordic% cd /usr/local/aeic/4.4/src
nordic% find . -name *.[123579] -exec fixman {} \;
nordic% cvs commit
```

This completes the update of all AEIC software.

The next step is to recompile the Iceworm software. For this, we will shut down the running nordic system. Again, we need to check out a new copy and compile and install it. The checkout only needs to be done once since /opt/iceworm/src is shared (bin,lib etc. have separate automounts though):

```
nordic% cd /opt/iceworm/src/
nordic% setenv CVSROOT /opt/iceworm/cvs
nordic% cvs checkout -d 4.4 src
nordic% cd 4.4
nordic% make Include
nordic% make install
```

The install turned up more man page trouble as well as a missing Makefile, so I fixed the latter (going back to 4.4u to do a *cvs add Makefile; cvs commit*) and ran *fixman* on all man pages[1]:

```
nordic% cd /opt/iceworm/src/4.4
nordic% find . -name *.[123579] -exec fixman {} \;
```

After this, it is worthwhile to clean up the remnants from the *fixman* process. I ran a *find -print* first to verify this was safe. If these old files are not cleaned up now they probably never will be[2].

1. This particular step will not be necessary for every Antelope upgrade. Invariably, a number of small issues come up each time a distribution is upgraded. It is very important to address them as they arise rather than deferring them. The upgrade task is a unique opportunity to reveal weaknesses in the software engineering and strategies for handling complexity. It is like a bright spotlight that illuminates a part of the operation, revealing the problems. Because developer time at AEIC is always limited, problems deferred are often ignored for a long time. Thus, deferring problems with the software distribution at this stage is tantamount to a commitment to mediocrity in the software base. That in turn guarantees that maintenance headaches in the future will take away more time from developers and operators. Also, by structurally weakening the software platform that is running now, these software problems ultimately limit what can be built on top of the existing software platform in the future.
2. Disorganization and messes in the software tree beg for bugs and mistakes. One might want to keep such old files around for rollback capability. However, the files also exist on backup tape and more importantly, in CVS. Rely on those for rollback; messes create problems. During this cleanup I discovered *orb2vdl* and its *makedate* executables registered with CVS. Executables do not belong in the CVS source distribution, so I cleaned this up also.

```
nordic% find . -name *.[123456789]- -exec rm {} \;
```

Next, I edited the rtexec.pf file for nordic to replace the multiple occurrences of 4.4u with 4.4 in environment variables and paths. After this I can restart acquisition on nordic.

At this point it is time to switch everybody in the lab to the new Antelope. In /usr/tools/setup/setenv, change

```
setenv ANTELOPE /opt/antelope/4.4u
```

to

```
setenv ANTELOPE /opt/antelope/4.4
```

Also in the same file, I removed tcl7.4tk4.0 from MANPATH.

It's usually good to test the new distribution now that everything is compiled. The real-time system appears to be running without problem. The other critical test is to make sure that *wormwatch*, the relocation utilities, and the alarm response utilities all run. The first problem encountered while testing this is that *wormwatch* relied on a compile step that is no longer appropriate. Because the *wormwatch* executable expects to auto-load subroutines from a data directory in /usr/local/aeic/4.4 rather than $ANTELOPE, I had constructed a makefile mechanism to bypass the standard Antelope compilation of .xwish "source" files. That amounted to constructing my own shell wrapper rather than letting antelope tools do the work for me:

```
wormwatch.sh : wormwatch.sh.ap
    perl -p -e "s@auto_path\s+DIR@auto_path $(DEST)@g" < wormwatch.sh.ap > $@
```

The antelope tools now install *awish* instead of *wish* in the .xwish headers, however; thus this mechanism is broken. Fortunately it is also no longer necessary since I've identified the necessity of the AEIC environment variable and installed it[1]. In wormwatch.sh.ap we can change

```
lappend auto_path DIR/data/tcl/library/wormwatch
```

to

```
lappend auto_path $env(AEIC)/data/tcl/library/wormwatch
```

This allows us to remove the custom wormwatch.sh Makefile target; move the source file wormwatch.sh.ap to the standard wormwatch.xwish; and remove all traces of customization in the latter. Finally,

---

1. This is a nice example of how investing in incremental quality improvements in the source code tree pays off over the course of months and years. It is a true investment: time and energy are necessary to 'do it right' whenever a shortcoming in the software base is discovered. Once again it is not possible to defer these 'payments' in a system like this. A commitment to the long-term growth of the technology must be made up front.

```
nordic% rm wormwatch.sh.ap
nordic% rm wormwatch.sh
nordic% cvs remove wormwatch.sh.ap
nordic% cvs remove wormwatch.sh  # Fix an old mistake (compile products shouldn't be in
CVS)
nordic% cvs add wormwatch.xwish
nordic% cvs commit
```

The *fktools* program allows a similar improvement, in that the makefile no longer has to substitute DIR in the auto_path assignment; instead the AEIC environment variable may be used. However, since *fktools* uses a nonstandard shell, we still need to generate fktools from fktools.sh. Unlike in the previous case, fktools.sh is a source-file, and needs to be added to the CVS distribution, after fktools.sh.ap and the corresponding makefile lines are removed. A *grepsrc* search on DIR and a find on *.sh.ap verifies that I have changed all occurrences of this problem in the aeic tree[1].

Returning to *wormwatch* and the attempt to relocate an event, we see that *dbchecker* is failing. First, /usr/local/seis_apps/dbchecker is compiled against Antelope 4.2u, as revealed by *ldd*(1). I fixed this. Second, *dbchecker_tool* does not use a proper makefile, thus it is still pointing to the old name for the *wish* executable. In fact, *dbchecker_tool* does not use a makefile at all[2]. Finishing these changes, the *wormwatch*->event-location->AEIC-release cascade now works.

The last step is to update the turnkey system-start file:

```
nordic% diff /etc/init.d/antelope /opt/antelope/4.4/data/system/S99_antelope
22,23c22,24
```

1. It's a very good idea, upon discovery of a shortcoming, to systematically hunt for and eliminate the same shortcoming throughout the entire source code tree. This is one of the keys to engineering success in AEIC development to date.
2. This is an opportunity to show the development and maintenance of a source-code tree. Had *dbchecker* and *dbchecker_tool* been part of the AEIC source-code tree, all of these version-skew problems and compilation problems would have been circumvented automatically. We had to discover and address the problems by hand because the utilities are not part of the engineered software base. Of course, a quick solution might be to put them into the /usr/local/aeic CVS structure. That was intentionally not done, however. The compilation problems and lack of makefile are only the first set of issues with these utilities. Many of the internal details of these programs also need to be changed to make them consistent with the whole Antelope/AEIC engineering platform. Some of these are at the design level. Many of the features of *dbchecker* and *dbchecker_tool* probably belong elsewhere in the analyst-review/checkin/quality-assurance flow. *dbchecker* and *dbchecker_tool* should probably not be split into two parts. Also, they fall in to a classic trap, that of addressing a problem by 'writing a script' or 'writing a GUI'. Often this is erroneously done without considering whether the functionality actually belongs in a script, or a GUI, or an executable of another sort, or a library; not to mention whether the functions actually all belong together at that one particular stage of the procedures. Addressing all these issues amounts to rewriting *dbchecker*. Finally, note that this isn't really to criticize *dbchecker* and *dbchecker_tool*; these were written as prototypes, scaffolding to be used while the whole system was being engineered. What is crucial is not to confuse the scaffolding with finished structure. The /usr/local/aeic distribution is for finished structure, or at the very least, code that may be re-engineered in the future but that is basically in keeping with AEIC design goals.

```
< @dirs = qw(/iwrun/dev/run) ;
< $user = "ppicker" ;
---
> @dirs = () ;  # fill in the actual directory of rtexec.pf, like below:
>             #       @dirs = qw(/export/home/rt) ;
> $user = "rt" ;  # fill in the actual user that runs rtexec
145c146
<       open ( LOG, "|/bin/logger -p alert" ) ;
---
>       open ( LOG, "|/usr/bin/logger -p alert" ) ;
nordic%
```

This change we can make be hand, switching logger to the /usr/bin version. I did this on nordic, ice, earlybird, fk, and marvin.

Now that nordic is ready, we need to propagate the upgrades to ice, earlybird, fk, and marvin.

```
nordic% rsh ice
ice% cd /opt/antelope/
ice% certainly_remove 4.3u
ice% certainly_remove perl_4.3u     # Clean up an old relic; should have been removed before
ice% certainly_remove perl
ice% certainly_remove tcltk8.3/
ice% cp -r /net/nordic/opt/antelope/{4.4,perl,tcltk8.3} .
ice% cp -r /net/nordic/opt/antelope/{data,doc} .
```

In response to a complaint from /usr/bin/cp:

```
ice% cd /opt/antelope/tcltk8.3/lib
ice% ln -s iwidgets3.0.1 iwidgets
```

The /usr/local/aeic distribution is already updated on ice due to rdist.

The iceworm code needs to be recompiled from the machine ice:

```
ice% cd /opt/iceworm/src/4.4
ice% make Include
ice% make install
```

This required changing the ownership of the whole /opt/iceworm distribution to kent:bdogs.

The *adsend2orb* executable actually runs out of /usr/local/suid_root. Therefore I did

```
ice:ppicker% cd /usr/local/suid_root/
# cp /opt/antelope/4.4/bin/adsend2orb .
```

I had to repeat this on nordic.

After this, I changed /iwrun/dev/run/rtexec.pf to have 4.4 instead of 4.4u in a number of locations. Also I changed /home/ppicker/setup/envrc and /usr/local/setup/setenv on ice to point to the correct Antelope version. At this point we can restart acquisition on ice, making sure to do so from a fresh shell.

As an aside, I removed the unnecessary /opt/antelope/antelope soft-link on nordic and the out-dated /opt/antelope/Changes and /opt/antelope/Release on nordic. I also removed /opt/antelope/dev on nordic, an old version of the contributed-code source that is no longer filed in that manner. We will need to remove tcl7.4tk4.0/ and tcltk8.0/ as soon as we remove /opt/antelope/4.3u.

Next upgrade fk, after shutting down the real-time system. Note that the "antelope" being removed is another stale soft-link; dev is the old contrib source code; and Changes, README, and Release date from Antelope 4.2u.

```
fk% cd /opt/antelope/
fk% certainly_remove 4.3u antelope README Changes Release dev doc perl_4.3u perl
tcltk8.3
fk% cp -r /net/nordic/opt/antelope/{4.4,perl,tcltk8.3,data,doc} .
fk% cd /opt/antelope/tcltk8.3/lib/
fk% ln -s iwidgets3.0.1/ iwidgets
fk% cd /opt/iceworm/src/4.4
fk% make Include
fk% make install
fk# vi /usr/tools/setup/setenv       # change ANTELOPE; MANPATH
fk# vi /etc/init.d/antelope          # verify previous change to /usr/bin/logger call
fk% vi /home/bbanddat/run/rtexec.pf # change all 4.4u occurrences to 4.4
```

Upgrade earlybird:

```
earlybird% cd /opt/antelope
earlybird% certainly_remove 4.3u Changes Release README antelope/ dev doc perl
perl_4.3u tcltk8.3
earlybird% cp -r /net/nordic/opt/antelope/{4.4,perl,tcltk8.3,data,doc} .
earlybird% cd /opt/antelope/tcltk8.3/lib
earlybird% ln -s iwidgets3.0.1 iwidgets
earlybird# chown -R kent:bdogs /opt/iceworm/{bin,lib,man,include,data}
earlybird% cd /opt/iceworm/src/4.4
earlybird% make Include
earlybird% make install
earlybird# vi /usr/tools/setup/setenv
earlybird# vi /etc/init.d/antelope
earlybird% vi /iwrun/op/run/rtexec.pf
earlybird% vi /home/ppicker/setup/pathrc
earlybird# cp /opt/antelope/4.4/bin/adsend2orb /usr/local/suid_root
```

The upgrade for marvin is similar, however the /usr/local/aeic software apparently needs to be copied over by hand:

```
marvin% cd /opt/antelope
marvin% certainly_remove 4.3u Changes Release README antelope/ dev doc perl perl_4.3u
tcltk8.3
marvin% cp -r /net/nordic/opt/antelope/{4.4,perl,tcltk8.3,data,doc} .
marvin% cd /opt/antelope/tcltk8.3/lib
marvin% ln -s iwidgets3.0.1 iwidgets
marvin# vi /usr/tools/setup/setenv
marvin# vi /etc/init.d/antelope
marvin% vi /home/uafarr/run/rtexec.pf
marvin:root 1 tcsh>> cd /usr/local/aeic
marvin:root 2 tcsh>> cp -r /net/nordic/opt/local/aeic/4.4 .
```

We have now upgraded all the local lab systems. Upgrades of alaskavsn (Pacific Geoscience Center, Victoria, British Columbia), alaskaseis (Hokkaido University, Hokkaido, Japan), and beam (Anchorage) are left as an exercise for the reader.