Filesystem Cleanliness Act of 1991
or
Filesystem Reorganization for Release 7.3

No Problem!

Athena Faculty Liaison

Everyone will know where everything is; it will be wonderful!

Athena Systems Support Supervisor

With this release, we are supporting three new platforms: RS/6000 with AIX, Macintosh with A/UX, and DECstations with Ultrix 4.2. One of the difficulties in Athenizing a vendor operating system is deciding where to put the Athena additions and replacements of vendor-supplied programs.

This topic is properly part of the VOSAP (vendor operating systems and platforms) discussion which we have not had time to start yet. Changes between the version originally sent around for comments and what was decided at the review are noted by changebars.

1 Motivation

By making this change now, there are no backward-compatibility requirements on the three new platforms. If we delay this change, we will have such problems on all of our platforms.

It should be possible to layer Athena software on a machine already running the vendor operating system. For this reason, Athena files should be added to a small number of places, not scattered across the entire operating system. Even for machines where Athena operations controls the system software, it is far easier for release-engineering to put together the release if the Athena changes are limited. This is also an important issue in re-integrating our changes when vendor updates are received.

2 Proposal

Athena additions are defined as anything we add to the vendor-supplied system, whether it was written here or imported.

1. Most Athena additions will go under **/usr/athena**, making this the mount point for the eventual layered Athena system. For third-party software, for the most part we will follow the supplier's suggested paths except that we insert ‘‘/athena’’ in the path after ‘‘/usr’’. However, all binaries will go in one place rather than creating **/usr/athena/new**, **/usr/athena/ucb**, etc.
    - User binaries will go in **/usr/athena/bin**.
    - Libraries will go in **/usr/athena/lib**.
    - Include files will go in **/usr/athena/include**.
    - Man pages will go in **/usr/athena/man**.
    - System binaries will go in **/usr/athena/etc**.
    - The imake configuration files will go in **/usr/athena/config**.
2. Some Athena additions must be on the root partition for use before the machine has come up on the network and mounted remote filesystems. Other Athena additions must be on the local disk because they are configuration files that may differ from workstation to workstation.
    - User binaries that must be local go in **/bin/athena**.

- System binaries that must be local go in the **athena** subdirectory of where the vendor puts such binaries (i.e. **/etc/athena** or **/sbin/athena**).
- System configuration files go in **/etc/athena**.

3. In some cases Athena must replace a vendor binary with an Athenized version. In most cases we will leave the vendor binary in place, and put our version in **/usr/athena**. The order of directories in the user's PATH will ensure that they get the proper version. We will also create a symlink with a different name for the vendor version (i.e. chsh.aix or lpr.ult) so that the user can still name the vendor program. For replaced programs that are invoked by absolute pathname (login, lpr, inetd, named, snmpd) we will rename the vendor version and leave it in the directory next to our own version.

3 Discussion

The following issues are examined with three audiences in mind: the student user in a public cluster, the faculty/staff person owning a workstation, and the software developer.

3.1 User Visibility

To avoid user visible changes in this release, we will put in full compatibility symlinks on the VAX, RT, and DECstation platforms. In addition, the RS/6000 and Macintosh platforms will have compatibility symlinks for user binaries, but not for developers.

At the time we release this software, we will notify users that these compatibility links will be removed a year from now, and give them pointers to documents for how to fix their dotfiles and Makefiles. The following discussion outlines changes that will be visible when we remove the compatibility links.

For people with dotfile customizations, there are three ways we expect users might lose. One is by using explicit paths, i.e. invoking `/usr/bin/X11/xterm` instead of just `xterm`. The most universal solution is not to use explicit paths and to relay on the PATH environment variable to find the programs. Another solution is to update the paths to the new locations, which will work on all supported platforms. We have yet to investigate the performance difference of using explicit paths versus using the PATH environment variable in dotfiles.

Another way people with customizations may lose is by setting their own PATH variable rather than by modifying the default one. People who just add to the end of `athena_path` will have no problems. People who set their own path will have to be educated to fix this.

The third way we expect that some people have have problems is by invoking features that are not on every platform. For example, people have been told to put `unlimit coredumpsize` in their dotfiles if they want to get coredumps. Other people have put `stty dec` in their dotfiles. These commands will not work on some of the new platforms. We must educate the users and provide sample dotfiles which have examples of machine-specific initializations. Something else we could do to combat all three of these problems is to provide a script which analyzes the user's dotfiles and suggests changes.

For programmers, there will be other visible changes. Existing binaries on the Vax and RT platforms will continue to work. We will also have compatibility symlinks on Vaxen and RTs for recompiling working code. However, to compile anything on other platforms additional flags will need to be given to `cc` to account for the move of include files and libraries. These flags will be forwards compatible on the Vaxen and RTs. We will be recommending to developers that they use `imake` to avoid these problems. Documentation for developers will be available by the time the release hits the field.

3.2 Backwards compatibility on existing platforms.

Compatibility for Vaxen, RTs, and DECstations will be forwards compatible symlinks rather than actually re-arranging the filesystems and putting in many backwards compatible symlinks. In this way, references to the old locations of files will still work, and through these symlinks references to the new locations will work as well. The required links are:

```
/usr/athena/bin -> /usr/athena
/usr/athena/include -> /usr/include
/usr/athena/etc -> /etc/athena
/usr/athena/third/bin -> /usr/new
/usr/athena/third/lib -> /usr/lib
/usr/athena/X11/bin -> /usr/bin/X11
/usr/athena/X11/lib -> /usr/lib/X11
/usr/athena/X11/include -> /usr/include/X11
```

The remainder of this proposal will not affect users on these platforms. It may not ever be worth re-arranging the VAX and RT platforms, although we will want to remove these symlinks from the DECstation in a year.

3.3 Man Pages

We have moved the Athena man pages, and mention the MANPATH environment variable. As of this writing, the only platform we have that does not support MANPATH is the DECstation. On the platform, we have our own version of `man` installed earlier in the search path than the Ultrix `man`. Keeping the man pages separate means that it is still possible to look at the original OS documentation, and our Athena layering will not interfere with what is an optional subset on many platforms.

We will need to add support to the standard dotfiles for an athena_manpath just like we do athena_path, and then educate users on how to use these.

3.4 X Integration

Where to put X was a difficult choice. One argument says that there is already a standard location for X binaries: **/usr/bin/X11**. But if the vendor puts their own version of X there, we have a conflict. And some third party software may depend on the vendor's version of X clients. So we put our X in a different place. Some vendors already put their X somewhere other than **/usr/bin/X11**, so this is not a major departure from standard practice. Any decision to separate X binaries from other Athena supplied binaries is somewhat arbitrary, so they will go in **/usr/athena/bin**.