ATK

Andrew Toolkit Consortium
School of Computer Science
Carnegie Mellon

The ATK View

Volume 1, Number 1
February, 1992

Welcome

Welcome to **The ATK View**, the newsletter of the Andrew Toolkit Consortium. With this issue we begin not only the newsletter, but the Consortium itself which was formed at the beginning of this year. In this issue we'll discuss the aims of the Consortium, our recent accomplishments, and our present projects. A special section in this issue will introduce you to the staff of the Consortium, all of whom are old hands in Andrew Toolkit development.

Directions: What is the Andrew Toolkit and what will it be?
Wilfred J. Hansen, Director

The Consortium supports the Andrew Toolkit and the Andrew Message System. Originally the Toolkit was envisioned as an extension of an earlier effort, edittext, an editor which offered typographic text: fonts, margins, styles, and so on. The extension was to allow embedding in text of arbitrary objects--drawings, rasters, spreadsheets, equations, and anything imaginable. It quickly became clear that the architecture we were inventing would suffice for embedding of any object in any other, and we adopted that goal. As applications of this architecture, we foresaw a new editor and the Andrew Message System, a Cadillac of electronic mail preparation, delivery, perusal, and management.

After initial development, the architecture attracted use by many researchers needing the combination of an architecture for embedding objects and ATK's full-featured text editing capability. Work on the toolkit was directed toward the needs of programmers developing applications by writing code in C.

With the formation of the Consortium, the principal members utilize ATK as part of the computing environment for diverse workers, many of whom are simply using the system and not developing code to extend it. Thus the focus of the Consortium for the immediate future is on addressing the needs of users rather than programmers.

One project, the preferences editor, will enable users to more readily tailor their environment to their own desires, often this means adapting it to behave more like some other editor. Another project, macro-extensibility, will also provide for user tailoring; in this case by writing scripts in a simple language.

I expect ATK to evolve into an environment where simple applications can be created without resort to programming in C. The screen image for the application will be created--and later revised--with a drawing editor. An extensive library of functions will be available for common application tasks like managing small data bases. And the connection between inputs on the screen image and the functions from the library will be made with the simple programming language. Many, perhaps most, users will be content with using the applications provided directly by ATK. Others, however, will find the flexibility required to build simple tools that have heretofore required advanced programming skills.

In fact the most important work of the Consortium will not be in development of new components or even extensions of the old. Rather, our critical function is to serve as the central point of concentration for the source code. As changes are made by workers throughout the world, we assemble them into the sources, check their operation on a variety of workstation types, and make them available in future distributions.

Upcoming

ATK Annual Meeting
    Tutorials: Thursday, June 25
    Annual Meeting: Evening of June 25
    Technical Conference: June 26

If you would like to
    make a presentation at the technical meeting
    schedule attendees at the tutorials
please send mail to susan+@andrew.cmu.edu.

June 15-19 Xhibition, San Jose (617 621-0600)

Recent Accomplishments

The work reported here has recently been completed and will be available on the CD-rom release.

CD-rom

To make ATK and AMS more widely available, we are producing a CDrom containing the Andrew sources and also executable binaries for the IBM RS/6000, DEC PMAX, Sun Sparc, HP700, and SCO for the i386. We hope to have the disc ready for distribution at X'Hibition in May. It has been interesting learning about making CDrom; we are working with software from Young Minds, Inc. to create the image prior to production by Disc Manufacturing Inc.

rdemo

The Remote Andrew Demo is a network service that allows users to interactively experiment with Andrew applications. All that's required is that the user be on the Internet and running X11. A simple ''finger'' command gets the demo started.

Originally offered in October, the service was taken down for some revamping and returned to service in January, shortly after the formation of the Consortium. It is running today on four hosts. From the many comments received, the demo has influenced users all over the globe. To the second week of February we served nearly 1,000 requests, with dozens from international sites including Germany, New Zealand, Switzerland, Australia, Japan, United Kingdom, Italy, and France. Several users have commented that the demo persuaded them to download, build, and install Andrew at their sites.

Some of the problems solved in creating the demo were security and performance issues, restricted versions of ATK applications, and the architecture of the demo session itself. (A document describing the details of the design of rdemo is available in the latest ATK release as the file rdemo/Design.d.)

Bibliography

Numerous papers have been written about ATK, AMS, and applications built on top of them. Now the Consortium has collected a number of these and made them available for anonymous ftp access from directory ./papers of emsworth.andrew.cmu.edu (128.2.30.62). The full bibliography of known papers is in that directory under the name BIBLIOGRAPHY; the collected papers are under names listed in the bibliography. (Also available via nationwide AFS as /afs/andrew.cmu.edu/itc/sm/releases/X.V11R5/ftp/papers.)

If you have written, or know of, other papers that belong in this collection, we would be delighted to hear from you.

Recent AMS Changes

John Myers and Christopher Newman, from CMU's Special Projects Group, have recently made speed enhancements and bug fixes to AMS. User visible changes include:

- Future dates in .AMS.prof files will be fixed automatically
- Bad pathnames in .AMS.prof files will be fixed automatically

Administrator visible changes:

- The .SubscriptionMap file may now be a symlink to another directory.
- An "other" directory is now supported for dropping master update hints.
- Master Update reconstructs should be less likely to interfere with hint adding.
- Netnews filing (nns) was rewritten to allow delayed filing of certain bboards.
- Added three new menu items: "This Message, Append To Raw File" and "Send/File Marked, Append To Raw File" for people who hate the text822 object when they're writing files. "This Message, Show Raw Body" is for people who want to see the underlying format of a non-text message. (Technically, it DOES undo the effect of a *top-level* Content-transfer-encoding header, but acts as if the Content-Type header were "text/plain". If you want to see what the message looks like WITHOUT undoing any top-level encoding, you should use the "append to raw file" menu.)
- You can now resend messages to a designated address from an init file (the parameter was previously ignored)

Bug fixes include:

- The "check new messages" routine formerly fetched the master update files. It now talks to a master update server which is running on bb2.andrew.cmu.edu. When other sites build AMS they will have to chose a server machine.
- All places which modify the .SubscriptionMap file were changed to follow a symlink. This allows the .SubscriptionMap file to be placed in a subdirectory, so the top level bboard volumes can be replicated read-only.
- If AMSNameSeparator is set for a given cell, AMS will now refuse to generate a full-name style address if that address is ambiguous in its cell.
- queuemail will drop mail in the test queues if the file /etc/UseTestQueues exists.
- recognize the print.printer preference.
- allow purging of messages when over quota.

Support for MIME (Multipurpose Internet Mail Extensions) Standard

MIME is the new proposed standard format for multimedia mail on the Internet. The old standard, RFC 822, specifies considerable detail about message headers, but expresses the message content, as flat ASCII text. MIME extends the format of message bodies to allow multi-part textual and non-textual message bodies to be represented and exchanged without loss of information. In particular, it is designed to provide facilities to include multiple objects in a single message, to represent body text in character sets other than US-ASCII, to represent formatted multi-font text messages, to represent non-textual material such as images and audio fragments, and generally to facilitate later extensions defining new types of Internet mail for use by cooperating mail agents.

Nathaniel Borenstein of Bellcore has implemented a MIME agent by modification of AMS. Messages will not only be able to recognize and display MIME-format mail, it will also generate either MIME-compliant mail or old Andrew-format mail, under user control. What this means in practice is that all the neat things you're used to doing with mail between Andrew users--formatted text, pictures, audio, and the like--will now interoperate with lots of non-Andrew systems as well. Andrew, however, provides a better integration and more pleasant user interface than any other MIME-compliant software now available.

Messages 8.5 can now read and properly handle MIME format messages of types "multipart", text/plain, text/richtext, and application/andrew-inset data. For other types, it will create a button object which, when pressed, passes the data to the "metamail" program, available freely from Bellcore. If metamail does not exist at the local site, Messages will offer to write the data to a file, undoing any of the standard transport-encodings that might have been used for mail transport.

When sending mail, AMS consults a new preference, *.mailsendingformat, with four options:

"ask" -- WHENEVER formatted mail is about to be sent out, regardless of any format-forcing codes in a user's .AMS_aliases file, give the user the choice of the old Andrew or the new MIME format. This is the default setting.

"andrew" -- behave as before, asking the user about sending formatted mail to non-local recipients, and using the old ATK data format whenever formatted mail is sent.

"mime" -- behave as before, asking the user about sending formatted mail to non-local recipients, but use the new MIME data format whenever formatted mail is sent.

"mime-force" -- Always use the MIME format, and don't even bother to ask about stripping to plain text. This should become an increasingly plausible option as time goes on, if MIME support becomes widespread, because the MIME format Andrew generates always begins with a readable text-only version of the message.

When Andrew writes MIME-format mail, it does so using the "multipart/alternative" construct. The first part is plain text, assuring widespread readability. Certain MIME types are, by default, converted into Andrew insets. In some cases this may be wrong -- e.g. if Andrew's alink object doesn't work on your machine, but your metamail has been configured to do something reasonable with audio & telephones. In such cases, you may use the "ForceMetamail" preference to give a comma separated list of MIME types that should always be passed to metamail rather than converted to Andrew, e.g. "*.ForceMetamail:foo,audio,bar".

Character sets. AMS assumes by default that you are using the US-ASCII character set and cannot tell from your font if this is not the case. However, AMS is sensitive to the "MM_CHARSET" environment variable (the same one used by metamail). If you are

using the Andrew 8-bit fonts, your .login should
setenv MM_CHARSET iso-8859-1
This will cause mail including 8-bit characters to be written out properly and will do the right thing with non-ASCII mail that you read with AMS.

Color Template Bug Fix

Formerly a bug in the style code prevented colors from being added to an existing document. For example, you could add color styles to a template, such as ctext.tpl, but when you went to apply them to a region of text, they would not show up. You can now have colors in your ctext!

Also, there is now a *Color* menu card in Messages that contains a set of popular colors. This menu card is defined in /usr/itc/released/lib/tpls/messages.tpl. You can create your own templates and use the ATKtemplatepath preference to make them accessible to your ATK applications.

Projects in Progress

preferences editor - The preference editor will obviate the need for editing preferences file, for memorizing bizarrely-named options and values, and for restarting applications before changed preferences are realized. The preference-editing application will know the names *and types* of all Andrew preferences. The type of a preference determines what values it can legally hold; for instance, the type of the Help.SearchPath preference is ‘‘list of directories,’’ while the type of the *.FontSize preference is ‘‘integer.’ This application will describe the type, purpose, and function of each preference, and permit the editing of the preference in a graphically-suitable style for the type of that preference. (For instance, the Help.SearchPath preference will look like a scrollable list of directory names *with completion* instead of a string like /foo/bar:/joe/fri/day:/and/so/on.) We expect to also add a ‘‘notify’’ button to this application. ‘‘Notify’’ will alert all presently-running ATK applications that one or more preferences have changed. The applications will be able to respond on the fly to the new preference settings.

The ATK programmer will also see a couple of changes. First, an extension to the Class (.ch file) language is expected. The names, types, defaults, and descriptions of preferences used by ATK classes will now be collected in the corresponding .ch files. At build time, this information will be placed in a central repository. Instead of calling the environ_* functions on string constants, as in

```
environ_GetProfileInt("fontsize", 12)   /* read my fontsize preference */
```

the programmer will now call something like

```
environ_GetProfileInt(pref_FontSize, prefdefault_FontSize)
```

where these constants are generated by the information in the .ch file.

The preference editor is in the early stages of design. We welcome your suggestions, and we'll keep you up to date on our progress.

macro-extensibility - ATK has long lacked a really useful macro facility, but that will change soon. Changes are being made which will allow Ness functions to be bound to keys or menu items "on the fly." The new macro definitions will be used immediately without restarting ez. When a macro has been defined by typing a sequence of keystrokes, the macro can be saved as a Ness function which calls the appropriate proctable functions. You can then modify the macro as you wish to make it more general or correct errors. A simple command is provided to append the new macro to an existing file.

Modula-3 - Although not the work of the Consortium, a related group is developing a Modula-3 interface for ATK. The goal is bi-directional: modula-3 objects will be able to invoke methods of ATK objects and ATK objects will be able to call methods of modula-3 objects. Each will be able to have pointers to the other. This magic is possible because the run-time representations were originally close enough that the ATK object representation could be modified sufficiently to match the modula-3 model. The result is that future development can be done in the modula-3 environment with automatic storage management, exceptions, an inter-module definition language sufficient to support strong typing in function calls, and numerous other pleasantries. (Unfortunately, the ATK portion of the system will still not have storage management.)

Changes to "class" - As part of the Modula-3 work, we contemplate eliminating from "class" four classprocedures which have been part of every object, but seldom used. (Note that ..._InitializeObject, ..._InitializeClass, and ..._FinalizeObject are not being changed.)

> *..._Initialize and ..._Finalize*: these were defined to perform the innards of ..._New and ..._Destroy, respectively. By calling them a client could convert arbitrary memory into an object and later detach it.
>
> *..._Allocate and ..._Deallocate*: these could be supplied by an object definition to give it control over memory allocation. In particular, for small objects a pool might be created instead of using malloc and free directly each time.

After elimination of these procedures, objects will have to be created and destroyed with ..._New and ..._Destroy, respectively. If elimination of these functions is a problem for any applications, please let us know.

Image - Work has begun on a color image inset based on the xloadimage application. At present, the image inset is primitive, providing only import of foreign image-files, display of images, and incorporation of images in ATK data streams. There are a few routines available for manipulating the whole of an image, but there is no editing interface.

Small data bases - In order to manage various small data bases such as bugs lists and member lists, we are developing a data base technology utilizing ADEW and Ness. The data base index will be in a simple text file and the details for any single entry can be found in associated files. Since the workings will be coded in Ness, it will be easily tailored to other data base tasks.

REXX functions - a suite of functions emulating some of the REXX string processing functions has been implemented in C for Ness. Unlike native Ness string functions the REXX functions force you to refer to positions in strings with integer values.

Staff

The Andrew Toolkit Staff would love the opportunity to meet each reader in person, especially those in California in March. A poor second is to introduce ourselves here. We have all had extensive experience in Andrew Toolkit development.

Fred Hansen, Director, joined the Information Technology Center in 1983, its first full year, and contributed much polishing to edittext, the precursor of ATK. He led the early stages of designing what later became ATK and was the inventor of the data/view separation that later became one of the hallmarks of ATK. Components designed and implemented by him in whole or in part include the lookz style editor, the raster image representation and viewer, and the Ness extension language. In earlier work, Hansen developed the first syntax directed editor, taught in various Computer Science departments, and developed operating systems and interactive applications for Three Rivers Computing Corporation. Off hours, Fred Hansen is a two dan go player, organizes tournaments for the Pittsburgh Go Club, and is active in the American Go Association.

Susan Straub has been with Carnegie Mellon since the inception of the ITC in 1983 as a staff member and student of Information Systems. She is the information contact for the Andrew Toolkit Consortium. She is responsible for release and distribution of ATK to the world and within the School of Computer Science. With the inception of the Andrew Toolkit Consortium, Susan will also handle the administrivia. Her current big project is the CDrom release.

Gary Keim has worked in the Andrew environment for more than five years. His earliest experiences were with the Base Environment 1 (BE1) toolkit, which he used while working for the Center for Education Computing in English (CECE) at Carnegie Mellon. He helped develop the **Comments** and **Notes** programs--hypertext applications used to facilitate writing tasks in a computer-saturated, networked world. Gary moved on to the ITC, where under Tom Peters he re-implemented the Bush filesystem browser and several supporting ATK classes. As a member of the ATK Group within the ITC--and now as a Consortium staffer--he has primary responsible for coordinating source updates and releases, as well as monitoring the ATK newsgroup. His current major project is the CDrom release. Gary is a fanatical hockey fan and is a right-winger on the CMU Hockey Club.

Bob Glickstein joined the Information Technology Center in 1987, while still a student at Carnegie Mellon. He earned a degree in computer science in 1988 and stayed with the ITC, creating and maintaining such Andrew components as ELI, FLAMES, AMS, parsec, yyhide, the Remote Andrew Demo, and many others. At present Bob is designing and implementing the preferences editor.

Rob Ryan is a recent graduate of Carnegie mellon. While still an undergraduate, he implemented several valuable new features: The CompChar package for typing in international characters; and motif style menubars, scrollbars, and dialog boxes. Rob has just about completed the macro-extensibility package.

Review - X Conference

Each newsletter features a review of some event of interest to readers. This month Fred Hansen reports on his trip to the X conference, Boston, 13-15 January:

The focus of the conference was the various technical papers. I’ve reviewed those of interest at the end of this article. My own attention was focussed on presenting ATK and understanding alternative systems.

For the tutorials on Monday, I attended the Interviews presentation by Mark Linton. Interviews is a very flexible toolkit written in C++. It stands at a somewhat lower level than ATK; you can program almost any desired appearance in Interviews, but you may have to do more work of assembling together the individual pieces. One of the really nice features is that most drawing is done to off-screen buffers with the final result copied to the screen. This makes screen updates much smoother.

On Monday and Wednesday evening, I described ATK and the Consortium at birds-of-a-feather sessions.

Wednesday afternoon I took an hour off and visited Chris Stone at the Object Management Group. This group is promulgating a standard for communication and data interchange among objects. The details will depend somewhat on applications, but there is provision for both dynamic communication across multiple platforms as well as persistence of objects in some form of data base. Communication is mediated by an Object Request Broker which translates data formats as necessary when traversing to alien platforms.

Thursday I devoted to the "XC++" working group. Originally this group was to create an interface to X from C++, but over time the emphasis has shifted to specifying an object oriented toolkit at a much more flexible level than the existing X toolkits. This effort is naturally taking Interviews as its starting point and merging in some of the major ideas from ATK since those systems offer many of the desired properties. During the day, the major presentation was from Andrew Palay, who managed the initial implementation of ATK. He presented a function layer similar to the ’view’ object in ATK.

The general nature of the papers at the technical conference was discussion of how to better implement or define various features of the X environment. As such, there was little of interest to those constructing applications on top of X in the way of ideas as to how to utilize X innovatively or advantageously.

A few of the highlights for developers at the Andrew Toolkit level of utilization of X:

Melarcame, Jeff, "Can you bet your life on X?" Using the X window system for command and control displays." The most interesting aspect of the presentation was the various forms of image and user interface in real-time command and control systems such as air traffic control. For instance, a separate processor is employed in some systems to display information acquired from radar as an overlay on other less dynamic information. Disappointingly, the paper says little about methods for ensuring the reliability of X; the basic approach is that the "software undergoes years of system integration test."

Raves, William, "A PostScript X server." To generate hardcopy from screen images this paper describes an X server that generates PostScript instead of rendering to a display. This paper describes only a feasibility demonstration, but ultimately the approach offers a means to get screen images with much higher quality and about a tenth the file size as a bitmap.

Smith, John Allen, "The multi-threaded X server." With the software described in this paper a user could operate in one window concurrently with image output in another; the system need not cease responding just because a lengthy image is rendered. The paper is not otherwise interesting to applications developers, but is mentioned here because it mentions the gamut of problems and solutions in implementing concurrent operations with threads. The references provide access to many papers useful for understanding this area.

Linton, Mark A., "Implementing resolution independence on top of the X window system." In X, coordinates depend on the resolution of the display. This was adequate when most displays were around 72 to the inch (one printers point per pixel). Now that displays with higher and higher resolution are arriving, this model is no longer satisfactory. The paper describes a C++ library in which applications draw on a real-world coordinate space and the operations are mapped to pixels by the library. Adequate performance was achieved. Just as with the cT system (a widely used educational computing environment from Sherwood and Sherwood at CMU) the font problem was solved by adjusting the pixel mapping to the physical size of the font. That is, a nominally 12 point font will force the rest of the image to be drawn assuming that the font is 12 points. Thus if the font is actually ten points high on the screen, the rest of the image will be 10/12’s of its nominal size. When the application is ported to a display with true 12 point fonts, the relations among image elements will be unchanged, although the entire image will be larger. The paper also discusses several issues concerned with rounding off to pixel positions.

Bartlett, Joel F., "Don't fidget with widgets, Draw!" This was one of the most interesting papers at the conference. The facility described is a graphics server that sits between an application and X and draws in response to a set of functions accessible from a Scheme language interpreter. For instance drawing a small black dot is done with

```
(fill-arc -5 -5 10 10 0 360 black)
```

In other words, the graphics package does not actually provide for the user to do drawing; it provides a set of primitives which an application can send, as text, to the graphics server. Each drawing primitive defines an object which can be given a name; object images can be repositioned or otherwise revised by issuing another drawing operation for the same name. A carefully designed event mechanism which reports events as text to the application simplifies construction of the application. Several applications are described in the paper, including an interactive weather map and a circuit routing package.

Rawal, Kuntal, "A macro facility for X." Macros provide for specifying repetitive actions. In the implementation described here, the macros are implemented in the server, so every application gets them automatically. Keystrokes and menu selections are easy to record in such a facility, but mouse hits are more problematic since their meaning is context dependent. Some attempts to solve the latter problem are described in the paper. ATK's im offers a similar facility, but it currently allows only one remembered macro at a time and it does nothing about context dependence for mouse hits.

Beged-Dov, Gabe, Ellis S. Cohen, "Implementing drag & drop for X11." One distinction between X applications and MacIntosh applications has been the lack of drag and drop in X. In the MacIntosh interface a file icon can be dragged onto an operator icon and the file will be appropriately treated: viewed, printed, discarded, or whatever. The X11 version has been implemented as a prototype under OSF/Motif(TM). Some controversy was evident in private discussions concerning the protocol since it has been defined to have the maximum flexibility and therefore the maximum implementation. Feedback while dragging is possible by modifying the icon dragged, the icon's origin, or the drop target the icon is currently over. The latter gives the most difficulty because it involves waking up the application supporting the drop site, even if the transition over it is coincidental with dragging to another intended destination. Consequently the protocol has provisions for various optimizations, each of which adds complexity to applications implementing the protocol. (This paper does not necessarily reflect the final protocol.)

Packard, Keith, "Using XTrap to help people with manual disabilities." Since some handicapped people have difficulty with the mouse, alternatives are useful. This paper describes an application 'hand' which interposes between input devices and X applications and translates user input events into events for the application according to a specification in an ad hoc language. This is a very general mechanism, but the paper is not clear as to how easy it is to use for keyboard entry of mouse positions.

Reader's Corner

The ATK Consortium is not a one-way street. We exist not only to contribute to ATK but also to coordinate the contributions of others. If you have articles of interest to our readers, please send them in.

And remember to prepare your technical articles for the Annual Meeting in June (see the **Upcoming** section above).

The Andrew Software

The Andrew Toolkit (ATK) is a portable user-interface toolkit that runs under X11. It provides a dynamically-loadable object-oriented environment wherein objects can be embedded in one-another. Thus, one could use our 'generic-object' editor (ez) to edit text that, in addition to containing multiple fonts, contains embedded raster images, spreadsheets, drawing editors, equations, simple animations, etc. These embedded objects could themselves contain other objects, including text. With the toolkit, programmers can create new objects that can be embedded as easily as those that come with the system. Many objects, including those mentioned above, along with a help system, a system monitoring tool (console), an editor based shell interface (typescript), and support for printing multi-media documents, are included in the release, making it useful to programmers and non-programmers alike.

The Andrew Message System(AMS) provides a multi-media interface to mail and bulletin-boards. AMS contains many advanced features including authentication, return receipts, automatic sorting of mail, vote collection and tabulation, enclosures, audit trails of related messages, and subscription management. It also provides a variety of interfaces that support ttys and low-function personal computers in addition to the high-function workstations.

Consortium Services and Offerings

For information about services and offerings of the Andrew Toolkit Consortium please contact us at:

Information Requests
ATK Consortium
Carnegie Mellon University
238 UCCB
4910 Forbes Avenue
Pittsburgh, PA 15213-3890
USA
phone: +1-412-268-6700
info-andrew-request@andrew.cmu.edu

We offer:

Full, distributing, and associate memberships
Source tape
Bibliography
Copies of papers
Videotapes
CDROM

Sources available

emsworth.andrew.cmu.edu (128.2.30.62) anonymous ftp
/afs/andrew.cmu.edu/itc/sm/releases/X.V11R5/ftp/... nationwide AFS
export.lcs.mit.edu (18.30.0.238) anonymous ftp
other X archive sites

Remote Andrew Demo service:

finger help@atk.itc.cmu.edu

News groups:

info-andrew+@andrew.cmu.edu (distribution list, send subscription requests to info-andrew-request@andrew.cmu.edu)
comp.soft-sys.andrew (same as info-andrew, but without ATK formatting)

Platforms

Andrew has been successfully used on (at least) these platforms:

IBM: RT AOS 3.4, RT AIX 2.2.1, RS/6000 AIX3.1, PS/2 AIX1.2
SUN: Sun3 3.5, Sun3 4.0, Sun4 4.0, Sun3 4.1, Sun4 4.1,
DEC: Vax Ultrix 3.1, Vax Ultrix 4.2, Vax BSD, DEC MIPS,
other: HP, SCO I386, SGI IRIX 4.0, Apollo, Macintosh II MacMach.

Send bug reports to
info-andrew-bugs@andrew.cmu.edu