Athena 6.2A Release:

Changes that Affect General Users And Program Developers

Kevin Cunningham
Athena Documentation Staff
12 June 1989

1. Introduction

In June of 1989, Project Athena installs the **6.2A** release of its workstation software on all public workstations (the release has been in test since early May). With this release, Athena adopts the latest available version of the X Window System (Version 11, Release 3) and incorporates other new software and bugfixes.

The most significant impact on general users will come from the new X release. This release includes new standardized approaches that are not compatible with past approaches, so ***several routines that used to work fine on Athena will no longer work in the new release***. Furthermore, ***users are expected to revise their own files or programs.*** In most cases, the corrective action needed is straightforward and need be done only once to permanently fix the problem.

Section 2 describes the X updates and the related problems users may experience. In particular, this section covers the most visible changes related to X11 Release 3 (font changes, cut-and-paste changes, revised locations of X development code, etc.)

The other sections of this document describe some new features in **6.2A** that may be of general interest to Athena's user/programmer community. Section 3 describes changes of interest to general users:

- A new set of file-deletion commands allow you to remove files in such a way that you can recover them later if you change your mind about getting rid of them (Section 3.1).
- The OLC Answers are now available both as part of the main release and from inside **olc**, the On-Line Consulting program (Section 3.2).
- The TeX and LaTeX family of text-processing tools are now available in the main release, providing an alternative to Scribe (Section 3.3).
- Several Scribe capabilities have been updated, and access to different versions of Scribe has been changed (Section 3.4).
- MATLAB mathematical modelling software is now available in a public locker (Section 3.5).
- Version 2 of the 2020 spreadsheet software is now available in a public locker (Section 3.6).
- The **discuss** conferencing system software is now available in the main release (Section 3.7).
- Bugs that made **compress** and **uncompress** lose information if you were over quota has been fixed (Section 3.8).

Section 4 describes changes of interest to developers and system hackers:

- A significantly improved version of FORTRAN, **hf77**, is now available on the IBM RT/PC (Section 4.1).

- The available C compilers have been updated (Section 4.2).
- The Scheme language has been removed from the main release (Section 4.3).
- The SMS database-management programs have been significantly revised, and are now known collectively as the MOIRA system (Section 4.4).
- Some files of interest to owners of *private* workstations have been changed (Section 4.5).

1.1 For More Information About 6.2A

If you are a systems hacker or someone else who would like to know the technical details of the changes made in Release **6.2A**, you can look at the **6.2A** release document written by Athena's System Development Group. To do so, use the **attach** command to attach the **release** locker, move to the directory **/mit/release/6.2**, and use the **viewdoc** program to examine the contents of the file **sysnotes.PS**:

```
athena% attach release
athena% cd /mit/release/6.2
athena% viewdoc sysnotes.PS &
```

This document contains an exhaustive list of the technical changes made to the Athena environment as part of Release **6.2A**, including lists of bug fixes and of changed system files.

2. Updates to X Window System

This section describes the highly-visible Athena changes associated with X11 Release 3:

- The set of fonts supported on Athena, and the naming conventions for identifying these fonts, has been radically revised. This may cause unexpected errors in a number of areas. See Section 2.3.
- Athena no longer supports the nonstandard way of sizing and positioning windows. You must use "-geometry" rather than "=". See Section 2.2.
- Cut-and-paste conventions for windows are slightly changed. This will create some unexpected behavior. See Section 2.4.
- The organization of the X Window System development environments has been completely revised, and the development environment, libraries, and include files have been relocated to predictably named lockers.

The following section summarizes the most likely user problems, with appropriate solutions. Later sections describe the changes to the release in a general way.

2.1 Problems and Solutions

Because the problems associated with the new release are considered the result of using now-outmoded methods (rather than from inherent bugs in X), ***users are expected to revise their own files or programs to correct the problems.*** In most cases, the corrective action needed is straightforward and need be done only once to permanently fix the problem.

The problems most likely to appear, and the appropriate solutions, are listed here:

- You may find that some call that used to create and size a new window automatically for you (e.g., one or more calls to **xterm** in your **.login.mine** or **.login** file) now produces an error message such as the following:

  `xterm:  bad command line option "=80x20+30+50"`

  `usage:  xterm [-help] [-display displayname] ...` *etc.*

  `Type xterm -help for a full description.`

  This means that the call to the X program uses an old form of sizing in which the dimensions prefixed with an equal sign (e.g., "=80x20+30+50"). This form of the call is no longer accepted in X11 Release 3. To fix the errors, you must edit the files from which the new windows are created (e.g., your **.login.mine** or **.login** file) so that the routine call uses the appropriate sizing convention, employing the -geometry switch (see Section 2.2).

  For example, if you are currently creating a new window with the following call:

```
xterm  =80x20+30+50  &
```

You must change it to:

```
xterm  -geometry  80x20+30+50  &
```

- In the new release, the fonts used by X have new names. This change may produce errors like the following if you have tailored your system to use special fonts:

  `Could not open font` *fontname*
  or
  `X server unable to find requested font '`*fontname***`'!.`**

  Errors like this mean that some routine is being called using an old form of font-naming, or looking for a font that is no longer supported. In many cases, the offending routine call will be in your own **.Xresources** file or another private file (e.g., **.login** or **.login.mine**, **.cshrc** or **.cshrc.mine**, **.uwmrc**, or **.Xdefaults**).

  If the problem lies in your own file, you need simply change the offending call to a form appropriate in the new release, as explained in Section 2.3.

  For xterm-type procedures, you can also temporarily get around the problem by typing the following command then restarting the program that produced the error:

  ```
  xset  fp+  /usr/lib/X11/fonts/old
  ```

  This provides only a temporary solution; you should change your font names to match the new ones as soon as possible.

  If the error-producing routine call is hardwired into an Athena program (rather than being adjustable in your call to that program), you should contact the person responsible for that code (or Project Athena), so that the callout can be revised.

- If you try to "cut" text from an Emacs buffer and "paste" it into another window on the screen, you may find that some other piece of text ends up being pasted in where you wanted the Emacs text. This is because some other piece of text is highlighted somewhere on the screen, and the cut-paste system in the new release of X always tries to paste the currently-highlighted text first, and only uses the most-recently-cut text if no text is highlighted.

  This behavior is slightly different from the previous release. To get the desired result, just make sure that no text is highlighted on the screen when you try to paste in the Emacs text to another window. This change does not affect wiping-and-yanking in Emacs buffers, but only cross-window cut-and-paste. See Section 2.4 for details.

2.2 "-Geometry" Switch Cannot Be Abbreviated Anymore

For some time now, the standard X calling sequence for specifying the size and location of a window has been of the form:

```
-geometry WidthxHeight+XOffsett+YOffset
```

An old X convention allowed you to specify these characteristics using an abbreviated form:

```
=WidthxHeight+XOffsett+YOffset
```

This latter form was supported on Athena for some time, even though it is no longer part of the X standards. In order to bring Athena up to X standards, ***this calling form is now no longer supported on Athena.***

If you have calls that use the old form, you must change them to use the new form in order to run your application. For example, suppose you have the following line in your **.login** (or **.login.mine**) file:

```
xterm =80x24+0+0 &
```

When you log in, the system will produce the following error message and fail to produce the new window:

```
xterm:  bad command line option "=80x24+0+0"
```

To fix the problem, change the line in your file to:

```
xterm -geometry 80x24+0+0 &
```

When you log in again, the problem will be fixed.

2.3 Font-Related Changes

The latest version of X addresses two major limitations in past versions of X:

- The set of fonts available on Athena in earlier releases has been replaced with a much more complete and regular set of fonts. This may cause problems for users whose programs or environment rely on fonts that are now unavailable (Section 2.3.1).
- The various ways to refer to fonts have been superseded by a single naming convention (Section 2.3.2).

These changes, and their implications for users, are described in the following sections.

2.3.1 New Set of Available Fonts

In all releases of X to date, there has been no standard, systematically collected set of available fonts. With X11R3, the arbitrary collection of fonts available in earlier releases has been replaced with a much more complete and regular set of fonts, contributed jointly by Adobe Systems and Digital. In addition, the names of these contributed fonts have all been made to conform to the new font-naming conventions (see Section 2.3.2).

With Release **6.2A**, only the fonts distributed with X11R3 will be visible to users by default. This means that any program that calls for particular fonts may fail because the desired fonts are no longer available on Athena, or because the call is not in a standard format.

For example, you may have tailored your Emacs to use a special font (e.g., by including a listing in your **.login** or **.Xresources** file). If you now try to call Emacs in **6.2A** but the specific font is no longer available, you will not be able to use Emacs, and the system will respond with an error message such as:

`X server unable to find requested font '`*fontname***`'!.`**

If you get such a message, you will need to change the callout. Of course, you will not be able to call your normal Emacs to help you edit the file! In this case, you can use a special call to Emacs:

```
emacs -font fixed
```

This will at least allow you to run Emacs so that you can edit the appropriate file to change the font callout in the **emacs** command. In general, you will simply need to change the font callout to identify a font actually available on the system -- and you will need to use the font-naming conventions outlined in Section 2.3.2.

Similarly, if you have any other commands or routines that fail because the fonts they refer to are no longer available, you will have to change the call to use an available font.

You may have to log out and log back in to have your changes take effect.

You can still use the old fonts (i.e., those previously available at Athena) if you want, but it will require explicit action to make them work for you. This should help motivate developers to actively fix applications that depend on non-standard names, particularly applications that have hard-coded font names.

2.3.2 New Font-Naming Conventions

The X Consortium is in the last stages of defining a standard font-naming convention. The naming convention divides a font name into 14 separate fields, each specifying one characteristic of the font that is expected to be of importance to a user (or application) when selecting a font.

A full font name conforming to the convention has the form:

```
-foundry-family-weight-slant-setwidth-style-pixelSize-pointSize-\
  Xresolution-Yresolution-spacing-averageWidth-registry-encoding
```

Note that a hyphen, -, is used as the field delimiter. In most cases only a subset of the fields should ever be specified explicitly, with the remaining fields wildcarded (e.g., using an asterisk, *).

When you specify a font name you need include only the minimum information that is crucial to the application. You are not required to specify a font-specification fully -- you can leave off later hyphens and matching wildcards (just make sure the fields you include are in the correct position in the list). For example, the following two specifications identify fonts acceptably (the first identifies a 12pt Helvetica, the second a 14pt Courier Bold

Oblique):

"*-helvetica-medium-r-*-*-*-120-*"
"*-courier-bold-o-*-*-*-140-*"

This general form is recommended:

"*-family-weight-slant-*-*-*-pointSize-*"

To see a list of the available fonts, enter the following command:

```
xlsfonts
```

This will display the complete list of fonts available on Athena, in the form appropriate for inclusion in font callouts.

Section (the Appendix) summarizes the font-name fields.

2.4 Cut-and-Paste Behavior Revised

Because of updates in X11R3, **xterm** and other applications that use the X Text widget (e.g., **xmh**, **xman**, etc.) now use a new cut-and-paste scheme involving two buffers. In **6.2A**, Emacs has not yet been revised to conform to this new strategy, so you may notice some odd behavior when you try to cut-and-paste from an Emacs window.

The new scheme uses the following general rules:

- When the user selects a region to *cut* (actually, the word "copy" might be more appropriate here), the system copies the text into a ***cut buffer***, and highlights the current text as the ***currently selected text***.
- When a user tries to *paste* text at the current active cursor position (if appropriate), the system first attempts to use whatever text is the ***currently selected text*** (i.e., whatever text is currently highlighted). If no text is currently selected, then (and only then) the system uses the content of the ***cut buffer***.

Emacs, however, still ignores the ***currently selected text*** and only uses text from the ***cut buffer***, whether or not there is any text currently highlighted on the screen. If you try to paste text from Emacs into another window and there is some text highlighted on the screen, the highlighted text will be pasted rather than the recently cut text.

The new scheme is the official X scheme, and the inconsistency noted above is due to the fact that Emacs has not been revised yet to conform to the new scheme. Future versions of Emacs will use the new scheme, but for now the system is somewhat inconsistent.

There are two workarounds for this problem:

- You can run the **xcutsel** client (type **man xcutsel** for more details).
- You can simply make sure that there is no ***currently selected text*** on the screen. You can clear the selection by clicking on the window currently owning

it; if you cannot find the window owning the selection, you can always make a new selection in any window you *can* reach, then clear *that* selection. Of course, since making a new selection places that text into the ***cut buffer***, you will have to "cut" the Emacs text again to place it into the ***cut buffer***.

It is possible to configure xterm (at the user or site level) to have only the pre-Release3 behavior, but this is not recommended.

2.5 New Homes for X Toolkit Files

In conjunction with the appearance of a new version of the X Window System, the organization of the X Window System development environments has been completely revised. In particular, the development environment, libraries, and include files for the X Window System have been relocated to predictably named lockers. Specifically:

- The X11R2 include files and libraries have been removed from the main release and placed in a locker named **X11R2**.
- The X11R3 include files and libraries now appear for the first time, in a locker named **X11R3**.
- The directory **/usr/new** has finally been purged of all X Window System clients. This may affect users whose **.login** or **.cshrc** files call **xclock**, **xinit**, etc., from **/usr/new** rather than **/usr/bin/X**.

This means you *must* explicitly attach one of these libraries when you want to compile your X programs. Whichever library you choose (**X11R2** or **X11R3**) is automatically attached as **/mit/x11**. If you attach **X11**, the most up-to-date library (currently **X11R3**) will automatically be attached as **/mit/x11**.

You usually need not have the development locker attached to *run* the procedure once it is compiled. Note, however, that there are a few exceptions (the X applications **plaid**, **xbiff**, and **xlogo** now require that the locker X11R3 be attached before you can run them).

See System Development's document on Release **6.2A** for more specific information on the new file reorganization (see Section 1.1).

The reorganization of the X development directories in some ways is meant to force developers to be explicit about their expectations. In particular, *developers should convert all applications using the X toolkit to use the X11R3 development sources as soon as possible*. With Release 3 of X11, the Toolkit intrinsics become stable with respect to the formal processes of the X consortium, so code is much more likely to be useable under future releases. (A document on converting widgets to X11R3 is available from Athena Systems Development.)

3. Changes of Interest to General Users

3.1 New Commands Allow Recovery of Deleted Files

As everyone knows, "**rm** is forever": when you use **rm** to remove a file, the file is erased from the system immediately and permanently. This behavior is usually what you want, but once in a while you may accidently remove a file you wanted to keep -- the manuscript for a paper that's due the next morning, for example, or part of your thesis! Unless you have already backed up the file yourself, there is no way to recover the erased file.

To help you avoid this problem, Project Athena has created a new set of file-deletion commands that let you remove a file in a way that permits you to recover the file (within a certain amount of time) before it is permanently eliminated from the system.

The new commands are summarized in the following table. (For more information about each command, including command-line switches, see the online man pages; for example, type **man delete**.)

Table 1. New File-Deletion Commands.

Command	**Function**
delete	Mark one or more files for permanent removal, making them invisible to the user but without actually erasing them from the disk (use **expunge** or **purge** to permanently erase a file marked for deletion)
undelete	Restore files marked for removal from current directory by **delete** (if not already expunged)
lsdel	List files marked for removal but not yet expunged
expunge	Permanently remove specific files marked for removal
purge	Permanently remove every file marked for removal in user's home directory and all subdirectories

For example, suppose you have a directory containing the following files:

```
Mail           thesis.mss   thesis.mss~
```

Because you are near your quota, you decide to remove the old version of your emacs file (the one ending with ~) to create some room. However, you accidently leave off the ~ from your command and thereby remove the newer version of the file from the directory:

```
athena% delete thesis.mss
athena% ls
Mail           thesis.mss~
```

If you had used **rm** to do this, you would not be able to recover the lost file, and would instead have to salvage what you could from the older file. However, because you used

delete instead of **rm**, you can recover the deleted file by using **undelete**:

```
athena% lsdel
thesis.mss
athena% undelete thesis.mss
athena% ls
Mail        thesis.mss      thesis.mss~
```

You can now remove the appropriate file, and even permanently eliminate it once you verify that you have marked the correct file for removal:

```
athena% delete thesis.mss~
athena% ls
Mail        thesis.mss
athena% lsdel
thesis.mss~
athena% expunge thesis.mss~
athena% lsdel
athena% ls
Mail        thesis.mss
```

Note that the **undelete** command will only retrieve files removed with the **delete** command -- it *cannot* retrieve files eliminated by **rm**. Thus, if you want to be able to recover files, you should always use **delete** to remove them. In addition, you cannot retrieve a deleted file that has been removed by **purge** or **expunge** (see Table 1).

You can set up your system so that **delete** is automatically used whenever you want to remove files, rather than **rm** or **rmdir**. Just put the following lines in your **.cshrc.mine** file:

```
alias rm delete -F
alias rmdir delete -D
```

Then when you type **rm** to get rid of a file or **rmdir** to get rid of a directory, **delete** will actually be used.

Because of the way **delete** works, deleting files does not actually lower the amount of quota you are using (each file is simply renamed to a form that is invisible to your normal work, specifically from *filename* to *.#filename*). To lower your used quota, you must fully remove the deleted files from your system by using the **purge** or **expunge** commands.

3.2 Revised OLC Capability

A revised version of the **olc** program offers a new menu option, **answers**. This option puts you in a browser to explore the OLC Answers, a collection of answers to the hundred (or so) most commonly asked questions about how to use Athena. These stock answers can be especially useful to you if no consultant is available to answer your question.

To see a list of all of the olc options, enter the command **olc** and then type a question mark, **?**, at the "olc>" prompt:

```
athena% olc
olc> ?
```

To select the OLC Answers, enter **answers** at the olc prompt:

```
olc> answers
```

The OLC Answers are also accessible from outside **olc**. Just enter **olc_answers** at your UNIX prompt:

```
athena% olc_answers
```

You do *not* have to attach a locker to access the OLC Answers (the stock answers were formerly available only in the **consult** locker).

3.3 TeX and LaTeX Provide Alternative to Scribe

The TeX and LaTeX text-formatting tools are now available as part of the Athena main release. These tools, formerly only available in the SIPB locker, provide an alternative to Scribe for creating and formatting text documents. TeX includes some improvements over Scribe, including better equation-handling and a good document previewer.

TeX itself is a text-formatting program that uses fairly complicated commands; however, LaTeX is a set of macros that make TeX easier to use. With LaTeX, you include simple commands in your document to produce complex formatting results (the LaTeX commands are converted into the more complex TeX expressions when you process the file with the **latex** command). LaTeX expressions are about as easy to learn as Scribe expressions.

The commands associated with this family of formatting tools are summarized in the table below. As with Scribe, you are expected to create a basic file that includes both your text and LaTeX commands *before* using the commands that create the actual formatted version of your text.

Table 2. LaTeX Commands.

Command	**Function**
latex	Process a file containing LaTeX commands (the file should have the suffix *.tex*), creating a formatted file (which will have the suffix *.dvi*)
tex	Process a file containing basic TeX commands (the file should have the suffix *.tex*), creating a formatted file (which will have the suffix *.dvi*); you should usually use **latex** rather than **tex**
xdvi *filename*.**dvi**	Create a separate window with options for previewing a formatted *.dvi* file
dvi2ps *filename*.**dvi**	Convert a formatted *.dvi* file to PostScript format (the resulting file will have the suffix *.PS*)

For more information about LaTeX, see the *Inessential Guide to LaTeX* available from SIPB. The Athena documentation group also plans to have an introductory guide to LaTeX ready later this Summer.

3.4 Scribe Revisions

Scribe has had new capabilities added and the location of various versions of Scribe software has been revised.

Some Scribe changes of note include the following:

- Scribe Thesis now allows users to include a copyright clause that gives MIT permission to copy and distribute the thesis (MIT requires that such a permission statement be included in any thesis copyrighted by the author). You include the clause by passing the argument 'GiveMITPermission' to the @ThesisFront command.

  This Scribe change was made possible through a change **mitfor.lib**, the standard frontpage library for Scribe Thesis. The previous version of **mitfor.lib** made no provision for the permission statement, and instead required users to hack a scribe database file to achieve the desired effect.

- Because most of the Athena-provided printers in public clusters are now PostScript printers, the default output device for Scribe is now *PostScript* (it used to be *LPTLN*). To use a non-PostScript output device, you should put a command of the form @Device(*device-type*) at the beginning of your .mss file.

- Support for landscape mode for printers has been added to the Scribe database. To use landscape mode, include the command @LibraryFile(Landscape) after the @Device command at the beginning of your .mss file. For example, to get landscape mode for a PostScript printer, include the lines:

  ```
  @Device(PostScript)
  @LibraryFile(Landscape)
  ```

- Support for grey color in Scribe documents has been added.

The latter new features are made possible by Scribe Version 6. This new version probably also solves other previous Scribe difficulties.

Scribe version 6.0 has been moved from the scribe6 locker into the main release as **/usr/athena/scribe6**. The existing scribe is likewise moved to **/usr/athena/scribe5**. Note that the command **/usr/athena/scribe** is the same as **/usr/athena/scribe5**.

3.5 MATLAB Available

MATLAB, a mathematical modelling package from MathWorks, Inc., is now available on a trial basis in the **courseware** locker. MatLab (*Mat*rix *Lab*oratory) is an interactive program that can help with scientific and engineering numerical applications.

To run MATLAB, just attach the courseware locker and run the program:

```
athena% attach courseware
athena% /mit/courseware/matlab/matlab
```

The current version of MATLAB runs only in an xterm window on Vax machines (i.e., *not* on IBM RT/PC), but MathWorks will be developing a version soon that will be able to run on both Athena machine types.

3.6 New Version of 20/20 Available

The latest version (Version 2) of Access Technology's 20/20 spreadsheet is now available on Project Athena (for Vax machines only). This version includes support for **xterm** tek mode graphics and Postscript output, in addition to more spreadsheet features. To use the new version, attach the appropriate locker and run the software:

```
attach 2020.r2
/mit/2020.r2/vaxbin/s2020
```

There is also a tutorial available. To run the tutorial, enter the following command (be sure the locker is attached, as above):

```
/mit/2020.r2/vaxbin/t2020
```

The current special key bindings for the xterm terminal type are described in the file **/mit/2020.r2/doc/keys.2020**.

No other documentation is presently available. Comments and/or bug reports should be sent to **eveham@ATHENA**.

3.7 Discuss Available in Main Release

The **discuss** conferencing system is now available in the main release. To enter the **discuss** system, simply enter the following:

```
athena% discuss
```

A copy of **discuss** is still available in the **sipb** locker, but you need not attach that locker to run the main release version. An article describing **discuss** is available in the **sipb** locker:

```
athena% attach sipb
athena% attach athenadoc
athena% /mit/athenadoc/viewdoc /mit/sipb/doc/discuss.PS
```

3.8 Compress and Uncompress Bugs Fixed

Bugs in **compress** and **uncompress** used to cause you to lose files if you tried to compress (or uncompress) when you were near or over quota. If there was not enough room on the system, the procedures would create .Z files of zero length, then delete the previous form of the file without having checked whether the new converted version had been successfully created. The result was the loss of your original file and a useless converted file.

This bug has been fixed. **compress** and **uncompress** will now give warnings and *not* try to convert the files, unless there is enough room on the system.

4. Changes of Interest to Programmers

4.1 New Robust Fortran Compiler on the RT/PC

hf77, a new FORTRAN compiler developed by IBM, is now available on the IBM RT/PC. **hf77** is significantly more robust than the FORTRAN compilers previously available for the RT, and also provides more explicit error messages.

The **f77** command is now symbolically linked to use **hf77**, which becomes the only FORTRAN compiler Athena officially supports on IBM machines. In addition, a new Fortran library has been installed.

The old version of **f77** is still available, as **/usr/bin/f77.old**. However, this old version is no longer supported by Athena.

The new **hf77** compiler has one known bug that may affect previously-developed applications: the compiler dumps core if you use variable names or subroutine names that are more than 11 characters long. (This comes out as "Termination Code 139" in the compilation stage.) FORTRAN convention has always required that variable names be shorter than 12 characters, but the FORTRAN compiler previously available on the RT did not enforce this requirement. With the new compiler, you must use variable names shorter than 12 characters long.

4.2 C Programmer Updates

The C compilers available in the Athena environment have been updated, and a workaround for a Berkeley 4.3 Unix error is now available. Both are explained below.

A new version of the High C compiler has been installed as the executable **/bin/hc2** on the RT. The advantage of this new compiler (the offically supported compiler for IBM ACIS) is that it compiles all of X -- if you are building the X Toolkit, the new compiler will definitely work, whereas the older compilers may not. The compiler changes are associated with recent IBM-provided kernel changes (including a new floating point scheme) for the RT/PC.

Note that the standard call **/bin/cc** will still point to **/bin/hc**, even though the latter has been de-supported. In addition, a symbolic link on VAX systems now links **/bin/pcc** to **/bin/cc**, for compatibility with the RT/PC environment.

Note to programmers: An error in Berkeley 4.3 Unix as originally implemented on both RT and VAX systems allowed a common C programming error (reading through a NULL pointer) to go undetected. The system still allows code with this error to fully compile and run, producing odd errors at runtime.

To make sure the code does not contain the NULL-pointer error, compile the code (e.g., with **cc**) using the **-Z** option. When you run the code now, it will automatically produce a core dump if the code tries to read through a NULL pointer.

The **-Z** option is a diagnostic tool only. You should be sure to recompile your code *without* using the **-Z** option once you have verified that your code contains no NULL-pointer errors. This is especially true if you are interested in porting your code to other environments: although code compiled with the **-Z** option will run on Athena, it may not run on non-Athena systems.

4.3 Scheme Removed From Release

To avoid having to maintain multiple versions of the Scheme programming language in the Athena environment, Project Athena has decided to remove the executables for Scheme from the main release. These executables are still available in the **scheme** locker, but Project Athena no longer supports Scheme as a standard release feature.

4.4 Revised SMS is MOIRA

The SMS programs have been significantly updated and renamed MOIRA, after the three fates in Greek mythology. The System Development guide lists the changes (see Section 1.1 of this document).

4.5 Private Workstation Owners: Take Note

Owners of Private Workstations should note that the automatic update procedure for workstations has been revised in ways that may affect you. In particular, if the variable PUBLIC is set to "true" in the file **/etc/rc.conf**, *all* configuration files on the workstation will be automatically replaced by standard versions of these files, and the executable **/etc/rc** will delete all of the files in the **/tmp** partition. In addition, the default file modes of the directores **/usr/tmp** and **/tmp** have been reset to 1777.

If you do *not* want the system to replace your configuration files (i.e., if you want to keep your workstation private), be sure to set the variable PUBLIC to "false".

For more information on these and other changes, see the System Development release document (see Section 1.1).

A. Font-Name Fields

As explained in Section 2.3.2, there is a new standard X convention for identifying fonts. A full font name conforming to the convention has the form:

```
-foundry-family-weight-slant-setwidth-style-pixelSize-pointSize-\
 Xresolution-Yresolution-spacing-averageWidth-registry-encoding
```

The 14 fields are as follows:

foundry the name or identifier of the digital type foundry that digitized and supplied the font data.

family a string that identifies the basic typographic style (e.g., "helvetica", "courier", etc.)

weight a string that identifies the nominal blackness of the font, according to the FOUNDRY's judgement (e.g., "medium", "bold", etc.)

slant a code-string that indicates the overall posture of the typeface design used in the font -- one of the following:

R	Roman	upright design
I	Italic	italic design, slanted clockwise
O	Oblique	obliqued upright design, slanted clockwise
RI	Reverse Italic	italic design, slanted counter-clockwise

setwidth a string that gives the nominal width per horizontal unit of the font, according to the FOUNDRY's judgment (e.g. "Normal", "Condensed", "Narrow", "Double Wide").

style a string that identifies additional style information not captured by other fields, but needed to uniquely identify the font in the FOUNDRY's judgement (e.g., "Serif", "Sans Serif", "Informal", "Decorated").

pixelSize an unsigned integer-string typographic metric in device pixels which gives the body size of the font.

pointSize an unsigned integer-string typographic metric in device-independent units which gives the body size for which the font was designed.

Xresolution an unsigned integer-string in pixels/dots per inch (dpi) which gives the horizontal resolution for which the font was designed.

Yresolution an unsigned integer-string in pixels/dots per inch (dpi) which gives the vertical resolution for which the font was designed.

spacing a code-string that indicates the escapement class of the font -- one of the following:

P	Proportional	proportional font
M	Monospace	a font whose logical character widths are constant, but no other restrictions apply.
C	CharCell	a monospaced font in which the glyphs can be modeled as boxes of identical width and height with no "ink" outside the box.

averageWidth an unsigned integer-string value giving the unweighted arithmetic mean width of all glyphs in the font, measured in 1/10th pixels.

registry a registered name that identifies the registration authority (i.e. international standards body) that owns the specified encoding.

encoding is a registered name that identifies the coded character set as defined by the specified registry.

Specifying the point size (in 10ths of a point) is generally preferred to specifying the pixel size as this allows the particular workstation configuration to direct the server to the appropriate fonts for the display device. (This feature is not currently in use at Athena, but may become useful in the future).

A more complete document on the font-naming conventions is available in the **x11dev** locker as **/mit/x11dev/src/doc/XLFD/xlfd.ps**

Table of Contents