The Athena Workstation 6.0 Release Notes

Bill Bryant
Documentation Group
Project Athena
Revision B
September 6, 1988

Table of Contents

1. The 6.0 Release -- A Brief Description

What follows is a list of changes and additions to the Athena Environment that might affect students, application developers, and other members of the user community. The changes are briefly described here; changes that require more explanation are noted in this list and described in more detail in the following sections of this document. A more technical description, the system notes, can be found in the *release* locker (/mit/release/6.0/sysnotes.PS)

These notes are fairly detailed, and many of the topics discussed may not really affect you. We discuss topics of general interest first, and topics of marginal interest later. Further information and updates will be made available in the locker *release*. You can access this locker as follows:

`athena%` **attach release**
`athena%` **cd /mit/release/6.0**
`athena%` **more README**

1.1 New Version of the X Window System

The X window system has been upgraded from **X Version 11, Release 1** to **X Version 11, Release 2** (**X11 R2**). X11 R2 includes both a new X server and a new **Xlib** library. X clients such as *xterm*, *xclock*, and *xload* have been rebuilt using the new Xlib. The release also comes with a more robust version of the **X toolkit**.

The protocol between the *X server* and *Xlib* has not changed. The programming interface between *X clients* and *Xlib* has not changed significantly. Certain variable declarations have been modified, and certain inactive arguments to functions are now activated. See Chapter 5 of this document for details.

X11 R2 changes two things that may affect users. X clients have new flags for specifying non-default displays and window geometries, and you no longer use a *~/.Xdefaults* file to specify client default values. See Chapter 4 for more information.

The *xterm* program's Tektronix emulation mode has had many bugs fixed, and should now be usable for most applications that require Tektronix emulation. **You no longer have to use the *xswitch* command to get an *xterm* that has a working Tektronix mode.**

The X10 Xlib has been removed from the release and is available in a separate locker. This is part of Project Athena's phasing out of X10.

This document's Chapter 5 describes in more detail the implications of X11 R2, and also outlines Project Athena's plans for the other versions of X that are currently supported. If you have developed or are in the process of developing an application that uses the X Window System, you must read this section.

1.2 Major Changes to Emacs Mouse Bindings

This release introduces Emacs Version 18.50. This Emacs has been recompiled with the X11 R2 Xlib library. The **Ctrl-spacebar** command sets the mark. Emacs handles its screen refreshing much better now. The protection mode on autosave files has been fixed as well.

Mouse functions have changed significantly. You can change the mouse functions back if you want to. Please refer to Chapter 2 for more information.

1.3 Zephyr Notification Service

This release includes a fully supported and activated **Zephyr Notification Service**. Names of some of the customization files have changed. See the Chapter 3 for more details.

1.4 New Function-Plotting Package - xmath

The *gnuplot* funtion-plotting package has been superseded by the program *xmath*, located in the locker of the same name (*/mit/xmath/xmath*). It does a variety of math functions, and can plot functions or data in X11 windows or Postscript-printable files.

1.5 dosread also Available on VS2000s

The */usr/ibm/dosread* command can now be used on VAXstation 2000 workstations, as well as IBM RT workstations, to read IBM PC format floppies. It cannot be used on VAXStation IIs because the command requires a 1.2 Megabyte floppy drive. Manual pages are available for it on the RT (**man dosread**).

1.6 Saber C Programming System

You can now use the **Saber C Programming System** on an RT workstation. (The command name is *saber*, with the program as an optional argument.) The RT uses Saber Version 1.2B. The VAXstation *saber* has been upgraded to Version 1.2C, which incorporates a number of bug fixes and enhancements.

VAXstations will also have access to Version 2.0 of Saber located on the *saber2* locker (*/mit/saber2/vaxbin/saber2*). This new version replaces the C syntax command interface with an interface like the one used in the Unix *dbx* debugging utility, and is much easier to use. *saber*'s online help describes the differences under the topic *release*. Some parts of the *~/.saberinit* files are incompatible between the two. If you customize it, you may need to use an ''*#ifdef SABER2_0*'' for those parts.

1.7 New Default C Compiler on the RT

The default C compiler (the one you get by typing **cc**) for the IBM PC/RT is now **hc**, rather than **pcc**. This compiler is more strict, efficient, and bug-free than the other. You might need to modify your source code, but it is still worthwhile to recompile old C programs. Object code from this is faster and more compact, but still compatible with **pcc**. A full description of the differences is found in the *Misc* directory of the *release* locker (*/mit/release/6.0/Misc/hcdoc*).

1.8 Increased Support for Makefile Dependency Generation

The 6.0 release includes a new programming utility **makedepend** (*/usr/athena/makedepend*). This utility uses a source file's preprocessor statements to determine the file's dependencies, then writes those dependencies to a Makefile.

See the manual page for *makedepend* (**man makedepend**).

1.9 Other New Commands

A PostScript version of */usr/ucb/vgrind*, */usr/athena/psgrind*, has been added to the system. *vgrind* is a program to format source files into pretty listings. Look at the man page for *vgrind*, as they are very similar (**man vgrind**) .

The command */bin/athena/hostinfo* has been added. Hostinfo asks the internet nameserver for a host's real name and network address. See its manual page for more information (**man hostinfo**).

1.10 Other Additions

A new version of *top* from Rice University has been installed.

A new version of */usr/new/jove* is installed.

Scribe v6.0 is installed as *scribe6*; Scribe v5.0 remains the default for this release, and now supports accents for the Helvetica and Courier fonts. The **Scribe** database for the Xerox 9700 printer has been modified to work better without Multics.

The **Kerberos** command for listing a user's tickets has been changed from *list_tkts* to *klist*. The *klist* command displays the active tickets in the way that *list_tkts* did when used with the *-l* option. For compatablility, *list_tkts* is linked to *klist*.

talk now uses the standard 4.3 port. The program */usr/ucb/talk.old* can be used to talk to 4.2 machines, though not 5.5T. During the release, old machines and new machines will not be able to *talk* to each other.

Network Information Protocol support has been added.

1.11 Enhancements to User-Oriented Programs

1.11.1 Enhancements to csh

The following changes have been made to the C-shell. Look at the *Misc* subdirectory of the *release* locker for more details (*/mit/release/6.0/Misc/csh.dvi*):

- */bin/csh* recognizes if you are trying to run a RT binary on a VAX or vice versa.
- The shell variable ‘‘hosttype’’ is set to ‘rt’ or ‘vax’ as appropriate.
- Keymaps can be dynamically allocated and assigned. New builtins have been added: list_keymap, set_keymap, create_keymap, delete_keymap. The old commands, set_edit_char and set_prefix_char, have been retained for backwards compatibility and affect the base and meta keymaps respectively.
- New editing functions were added: delete_eol, emacs_twiddle, upcase_word, downcase_word, capitalize_word, and no_op. ^K is set to delete_eol instead of delete_search, a backward incompatibility.
- The prompt string now expands various %-sequences: For example, the string "<juser@%m> {%d}\n%h% " generates a prompt like this:

```
<jruser@SELENE> {/mit/jruser}
137%
```

- The shell variable prompt2 controls the prompt printed during interactive "while" and "for" loops.
- The *history* list is now time stamped; use the *-t* flag to see this.
- If the alias *precmd* is set, then that alias will be executed before the prompt is printed. This is useful for people who like to have cycling prompts.
- Command completion has been enhanced. During rehashing, a command table gets built for the purpose of command completion. This speeds up command completion into something usable.

1.11.2 New Options to attach and detach

The *-n* option does not attempt to do mount time authentication for NFS mounts, unlike the default case. This replaces many uses of *-r*. With *detach*, the flag indicates making no attempt to remove authentication.

The *-z* option does not attempt to subscribe to Zephyr notification about the file server, unlike the default case. With *detach*, the flag indicates making no attempt to remove subscriptions.

1.11.3 Console Output to an xterm Window

VAXstations now support redirection of console output to a window using the *-C* option to *xterm*. This sends error messages to that window, rather than giving you black-bar messages.

1.12 Bug Fixes

Various device support improvements for the VS2000 are installed. The floppy drive device */dev/floppy* has been linked to the correct output device on all types of hardware.

The X window dump program *xdpr* now works.

MicroVAX emulation of VAX floating point instructions and *matchc* string comparison have been fixed.

/usr/bin/lint now handles ENUM's correctly.

/usr/bin/X/emacs, */bin/cp*, and */bin/mv* have been modified to check for write errors more carefully, in conjunction with the kernel change described above. This makes it more difficult to lose files if you accidently go over quota.

The program */etc/telnetd* has been fixed so that PCs can successfully *telnet* to Athena hosts.

/usr/bin/ranlib is now careful about handling clock skew across NFS; the previous hack to handle this in */bin/ld* has been left in for this release, but will be removed a future time.

A special note to those who have written shell scripts that use the */usr/bin/egrep* utility: a period (.) no longer matches a newline.

/bin/chgrp through NFS now works.

1.13 Support Removed for RFS and MDQS

Client support for RFS has been removed. Support for MDQS, RFS, and *userinfod* has been removed from */etc/rc*.

1.14 Enhancements to System Management Programs

The */etc/athena/newsyslog* program covers more system logs, and operates under the control of a configuration file, *newsyslog.conf*

A single message has been placed in */usr/msgs* indicating that the */usr/ucb/msgs* command is obsolete. Private clusters can still choose to turn on *msgs* by replacing a symbolic link with a hard file.

SLIP support was added to *\/etc/pstat*.

If the workstation is public, then a new */usr/lib/sendmail.cf* file will be reinstalled from */srvd* at every boot.

1.15 The inetd Program and access_on/off

If you use a private workstation, you should be aware there is a new way to enable remote access. The */etc/inetd* now brings up the workstation in *access_off* mode, but you can use *inetd*'s *-n* flag to force the workstation into *access_on* mode. It is no longer appropriate to include a *kill -1* of *inetd* in an *activate* script, because *inetd* will merely restart -- it won't switch to *access_on* mode.

You can still put your workstation into *access_on* mode by including the command in the */etc/athena/deactivate.local* script. Note that this does not insure absolutely that a workstation will be accessible remotely after a reboot sequence. Should the workstation be unable to complete the first *activate/deactivate* cycle, it will remain unavailable.

1.16 New Libraries

The libraries described below are local to Project Athena. If you use their functions when developing an application, please be aware that your code is not portable to other sites unless you send the source to these libraries along with the rest of your application. You can consider the source-language version of these libraries as part of your own set of programs. They are owned and copyrighted by M.I.T., but they come with blanket permission for royalty-free reuse.

1.16.1 The Access Control List Library

The Access Control List (ACL) library is now available for developers as *acl* (*/usr/athena/lib/libacl.a*). The functions in this library let you add a level of authorization to your applications.

1.16.2 The SS User Interface Library

The SS user interface library is now available for developers as *ss* (*/usr/athena/lib/libss.a*).

1.16.3 The ET Error Table Library

The Error Table (ET) library is now available for developers as *com_err* (*/usr/athena/lib/libcom_err.a*).

2. Major Mouse Change in Emacs!

The 6.0 release includes major changes to the way you use the mouse in the *emacs* editor so that *emacs* now behaves like a "standard" X toolkit application. If you use the mouse a lot when using *emacs*, get ready for a major readjustment.

If you'd rather keep the old mouse bindings instead of learning the new ones, add the following line to the *.emacs* file in your home directory:

(x-set-x10-mouse-bindings)

If you get caught in an *emacs* session with the new bindings and you want to change them in midsession, press *Esc-x* or *Meta-x*, then type the following:

x-set-x10-mouse-bindings

2.1 The New Bindings

In previous versions of Emacs, you could use the mouse to reposition the cursor by moving the mouse cursor to the desired location, then pressing the mouse's center button. If you were using an RT workstation, you pressed both mouse buttons simultaneously. **THIS NO LONGER WORKS!** Instead of repositioning the mouse, the center button inserts text from the *cut-and-paste* buffer.

Now you must use the mouse's **left** button to reposition the cursor. Move the mouse cursor to the place where you want to move the regular cursor, then press the left button.

2.2 Cutting and Pasting Text

The text-*cutting* operation in *emacs* has not changed, Position the regular cursor at the beginning of the text block you want to cut; move the mouse cursor to the end of the text block; then press and hold the *Ctrl* key, and press the mouse's **center** button.

To mark and paste text in *emacs*,proceed as follows:

1. Move the mouse cursor to the beginning of the text you want to mark, and press the **left** button to reposition the regular cursor.
2. Move the mouse cursor to the end of the text and press the mouse's **right** button. You have now copied the text block into the paste buffer.
3. Move the regular cursor to the place where you want to insert the paste buffer's contents, then press the **center** button (or both buttons if you are using an RT workstation). Emacs pastes the text into the file.

Other mouse functions remain the same. Read the Emacs manual page (**man emacs**) for details.

3. The Zephyr Notification Service

The ***Zephyr*** notification service provides a means of sending small quantities of timely information from one point on the network to another. You can use it to communicate with another workstation user on the other side of campus. File server machines can use it to notify their users about pending maintanance work. When you get a message sent via the Zephyr system, the message gets displayed in a *windowgram* on your workstation's screen.

The following is a list of Zephyr client programs and their functions:

zctl	The Zephyr subscription control program. Use this program to subscribe various message types. It also subsumes the functionality of *zhide* and *zflush*. For more information, type **man zctl**.
zlocate	Use this command to see if and where a user is currently logged in to an Athena workstation.
zwrite	Send a Zephyr message to another user. Type **man zwrite** for more information.
zleave	Use this command to set an alarm for when you have to leave. Type **man zleave** for more information.
zaway	Use this command to notify message senders that you are away from your terminal. Type **man zaway** for more information.
znol	This program waits for designated people to login and sends you a message when they do. You get to designate the people, of course. Type **man znol** for more information.
zwgc	Not really a command, but rather the name of the program responsible for displaying Zephyr notices on your screen. If you are interested in customizing the look of your Zephyr notices, type **man zwgc** for more information.

4. User-Visible Changes to the X Window System

X11 R2 changes two things that may affect users. X clients have new flags for specifying non-default displays and window geometries, and you no longer use a *.Xdefaults* file to specify client default values.

4.1 New X Client Flags

X clients, such as *xterm* and *xclock*, have two new flags, *-display* and *-geometry*.

In previous releases of X, you could tell a client program to display itself on an alternate display by including the display's name in the client's command line:

```
host% xclock ariadne:0
```

This no longer works. If you want to display a client on an alternate display, **you must use the *-display* flag**:

```
host% xclock -display ariadne:0
```

If you have written any shell scripts that start X clients on non-default displays, you need to update them to include the *-display* flag.

The *-geometry* flag isn't as critical an issue at this juncture. In previous releases of X, you could specify the size and position of a client's window like this:

```
host% xclock =70x70+1+1
```

This still works. You can, however, use the *-geometry* flag to do the same thing:

```
host% xclock -geometry 70x70+1+1
```

Even though the old specification syntax still works, we recommend that you add the *-geometry* flag to any shell script you have that specifies a client's size and position. Future release of X may **not** suppport the old specification syntax.

4.2 Specifying X Defaults in Release 2

In previous release of X, you could create a *~/.Xdefaults* file in your home directory to customize aspects of a client's window, such as border width and background appearance. The *.Xdefaults* file was a holdover from the days of *X Version 10*.

The X11 R2 server no longer recognizes *~/.Xdefaults* files. Instead it looks for a *~/.Xresources* file in your home directory. Not only has the filename changed, but the syntax within the file has changed as well. The new syntax is described in the new manual page for X, under the RESOURCES section. (Type **man X**.) Individual programs also list possible resources in their own man pages.

The 6.0 workstation provides a program that can convert simple *~/.Xdefaults* files to *~/.Xresources* files. The program, called *xdef2res*, is rudimentary and may not be able to generate the correct *~/.Xresources* files from complicated *~/.Xdefaults*. The *xdef2res* program and other information about X resources are documented further in the X subdirectory of the *release* locker (*/mit/release/6.0/X*).

5. Developer's Notes on X at Athena

The 6.0 workstation release introduces to the Athena environment a new version of the X window system: **X Version 11, Release 2** (**X11 R2**). Anyone who has developed an X application should consider updating it to use the new X libraries.

In addition to X11 R2, Project Athena supports, to varying degrees,three versions of the X window system:

1. **X Version 10** was developed before the X window system became an industry standard. It was the only version of X available before June, 1987. The introduction of *X Version 11* has made X10 obsolete, and Project Athena is in the process of phasing it out.
2. **X Version 11, Beta Release** was a prerelease of X11 that was used at Project Athena during the summer of 1987. If you developed and built X11 applications between June, 1987 and September, 1987, you used the Beta release of X11.
3. **X Version 11, Release 1** was the first official release of X Version 11. This release was used at Project Athena from September, 1987, through June, 1988. If you have developed and built X11 applications since September, 1987, you used X11, Release 1.

5.1 X11 R2

X11 R2 uses the same protocol as that used in X11 R1. Since this client-server interface has remained unchanged, binaries that were loaded with X11 R1 functions can successfully run with the X11 R2 server.

5.1.1 Converting X11 R1 Applications to X11 R2

Some of the header files used by *Xlib* functions have been changed. This means that you have to recompile X applications using the X11 R2 header files **before** you reload them with the R2 Xlib. Every X-using module must be recompiled with the R2 header files.

The R1 Xlib had a number of functions that were called with unused (ignored) arguments. The R2 Xlib has activated many of these arguments. If your R1 application has supplied values for arguments of this sort, be aware that the R2 X server will now act upon those arguments, perhaps causing the application to behave unexpectedly.

Be prepared for the possibliliy of finding minor bugs in your application when you reload it with the new library. Project Athena's Steve Ellis is compiling a list of applications that generated bugs when loaded with the R2 libraries. His username is *ellis* and he can be reached at x3-1313.

Athena now provides *lint* libraries for X applications. If your application passes through *lint* and through *saber* without generating any errors or warnings, chances are the code will be easy to port to other systems.

5.2 Support for the Other Versions of X

Athena's policy for support of the other versions of X is discussed in the following paragraphs.

5.2.1 X Version 10

This version of X is obsolete and is gradually being phased out at Project Athena, starting now.

In the 6.0 release, the libraries used to build X10 applications are no longer part of the standard system. To build an X10 application, you must first attach the **x10** NFS locker before you can access the libraries:

`host%` **attach x10**

As has been the case since September, 1987, you will be able to run X10 binaries by using the *xswitch* command to replace a workstation's X11 server with an X10 server. Note however that this modicum of X10 support is temporary. **Project Athena will remove *xswitch* and the X10 server from the standard system in January, 1989.**

With the release of the 6.0 workstation system, Project Athena introduces an X10 translator program that may be used to run certain X10 applications on an X11 server. This program, called **x10tox11**, creates a pseudo-server that converts X10 library calls into calls that an X11 server can understand. Use the translator as follows:

`host%` **x10tox11** ***application_name***

where *application_name* specifies the name of the X10 application you want to run.

Because of its rudimentary nature, the translator may not work for all X10 applications. Project Athena does not plan to extend its capability. If Project Athena receives an improved version of the translator from another party, we will add it to our system.

If you use an X10 application that you would like to run in the Athena environment after January, 1989, you must modify it to use the X Version 11 system. The conversion of an application from X10 to X11 takes a considerable amount of effort because the protocol is entirely different. If you are contemplating a conversion, refer to the document *A guide to Converting X10 to X11 Xlib Calls*, found in */mit/release/6.0/X/X10-correspond.doc*.

5.2.2 X Version 11, Beta Release

If you compiled and loaded an X11 application between June and September, 1987, the resulting binary calls routines from the Beta Release of X11. Binaries of this type have been running since September, 1987, and will continue to run until January, 1989. At that time, Project Athena will remove the Beta Release compatibility feature that allows such applications to run in the Athena Environment. If you use applications that were built under the Beta release, you should rebuild them using the X11 R2 libraries before the January, 1989 deadline.

You can check if your application uses the Beta Release of X by running it on the machine *Selene.MIT.EDU*, provided by the Project Athena Systems Development Group. This workstation cannot run Beta applications successfully. To test your program, simply run it with the *DISPLAY* set to "Selene.MIT.EDU:0". *Selene*'s X server will complain about an X protocol mismatch and refuse to connect to Beta applications (other applications will just fail due to an access violation). If your application fails due to the mismatched protocol, you need to rebuild it with the current version of the X libraries. This service is available for a limited time only, so make use of it as soon as possible.

5.2.3 X Version 11, Release 1

If you compiled and loaded an X11 application between September, 1987 and June, 1988, the resulting binary calls routines from the X11, Release 1 libraries. Your application will be able to run on X11, Release 2 servers through the 1988-1989 academic year.

Plan to recompile and reload the application within the academic year. The X11, Release 2 library provides several bug fixes, and Project Athena will be able support your reloaded application for a longer period of time.