Athena 6.3B Release:

Changes that Affect General Users
And Program Developers

Kevin Cunningham
Andrew Borthwick-Leslie
Athena Documentation Staff
23 August 1989

Table of Contents

1. Introduction

In August of 1989, Project Athena installs the **6.3B** release of its workstation software on all public workstations. Very significant changes have been made to both the appearance and operation of logging in and out. The changes consist of the installation of *xlogin*, a part of *xdm* (X display manager), and a rearrangement of the 'dot files' system (especially .login.mine). For the average user, once the initial 'Oh no, not again' has escaped her lips, the adjustment will be quick and painless. These changes are covered in Chapter 2 XDM and Dotfile Changes but depending on what type of a user you are, you need not read all of it:

- If you have never customized your logins (if you don't use *.login.mine* or haven't altered *.login*) you will find very little difference. Read at least section 2.2 *The new "login" window*, section 2.3 *The "console" window*, and section 2.4 *Logging out and the 'logout' button* for an overview. The details aren't necessary.
- If you enjoy the flexibility of customizing your logins, should read Chapter 2 *XDM and Dotfile Changes*, section 2.5 explains how to convert your old customizations to the 6.3B scheme. We hope this will help you quickly adjust.
- The truly ambitious hacker can take advantage of **xdm**'s ability to allow users to completely alter the way their sessions are run, and should read the document that completely describes the **xdm** and related 'dotfile' changes in the document described in the next paragraph.

For a very thorough description of the new system read Kevin Cunningham's document **"Dotfiles" : configuring your session in Release 6.3**. To view that document, use the **attach** command to attach the **athenadoc** locker, move to the directory **/mit/athenadoc**, and use the **viewdoc** program to examine the contents of the file **dotfiles.PS**:

```
athena% attach athenadoc
athena% cd /mit/athenadoc
athena% viewdoc dotfiles.PS &
```

Also, hard copies should be available in public clusters.

The other sections of this document describe some new features in 6.3B that are of general interest to Athena's user/programmer community. Chapter 3 describes the following changes of interest to general users:

- The Motif window manger, mwm, is now available in the main release. MWM WILL BE THE DEFAULT WINDOW MANAGER STARTING THIS JANUARY. (Section 3.1)
- Athena's documentation will be changing from the current system of on-line *essential* and *more* documents to **OLH** (on-line help) which can be accessed by typing *help*. (Section 3.2)
- **emacs** has new switches and new ways of describing old ones (some of which will now break) (Section 3.3)

- **xmh**, the X interface mail system has some friendly improvements. (Section 3.5)
- There's a handy new way to find out you don't have any file storage left: **xquota**. (Section 3.6)
- We have updated to Scribe Version 6, and Scribe Version 7 is available in a locker. (Section 3.4)

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

- A new version of Saber-C has been installed. (Section 4.1)
- The Andrew file system (AFS) is now available. (Section 4.2)
- x10 binaries have been moved to a locker. (Section 4.3)
- There are new versions of **make** and **rcs**.

1.1 For more information about 6.3B

If you are a systems hacker or someone else who would like to know the technical details of the changes made in Release **6.3B**, you can look at the **6.3B** 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, and use the **xdvi** program (the document is written in *LaTeX* format, and *xdvi* is the *LaTeX* previewer) to examine the contents of the file **/mit/release/6.3/rel6-3.dvi**:

```
athena% attach release
athena% xdvi /mit/release/6.3/rel6-3.dvi &
```

/mit/release/6.3/rel6-3.dvi contains an exhaustive list of the technical changes made to the Athena environment as part of Release **6.3B**, including lists of bug fixes and of changed system files.

2. XDM and Dotfile Changes

With Athena Release 6.3B, users will see several significant changes related to a change in how "user sessions" are defined:

- There is a new initial "login" interface on all workstations, offering several new options in addition to Username and Password.
- Athena renamed a number of configuration files *(also known as "dotfiles")* in every user's home directory, and moved new configuration files into these directories to work in place of the old files.

These changes affect every Athena user, and require many users to take action to adjust to the new system (see especially the Section 2.5 on converting to 6.3B).

The reasons for the changes, details of the changes, and what users should do to adjust to the changes are described below. For a very thorough description of the new system read Kevin Cunningham's document *'Dotfiles': Customizing Your Session in Release 6.3* in the athenadoc locker.

2.0.1 Why the changes?

The changes were made partly to correct some minor problems with earlier versions, but primarily the new organization of files addresses the fact that a "window session" in the workstation world is different from a "login session" in a single-window world (a single-window world might be the result of dialing-up or remote logging-in across the network).

Athena tried to address the inherent difference between "login" and "window" sessions by bending an old "login" paradigm to serve the new "window" needs. This compromise solution created awkward and unsolvable problems for many users. Recently, the X Consortium (which provides standards for X-window-based systems) began work on a more appropriate approach to "window" sessions, making adjustments to the workstation login window and user configuration files necessary and appropriate.

One other principle underlies the new set of files: Athena should avoid changing user files whenever possible. Until now, system activity that needed to be run when any user logs in was put into the user's home directory (~/.cshrc & ~/.login). When changes were needed in this activity, Athena had to tamper with user files. By moving all such system activities out of user directories and into system configuration files, Athena can make minor changes to the configuration files without having to change user files every time.

2.0.2 What got changed?

Any files with the names listed below were moved aside, by appending the suffix ".prefall89" to each filename (you probably only had the first two to four files):

File named:	Renamed to:
~/.cshrc	**~/.cshrc.prefall89**
~/.login	**~/.login.prefall89**
~/.cshrc.mine	**~/.cshrc.mine.prefall89**
~/.login.mine	**~/.login.mine.prefall89**
~/.xsession	**~/.xsession.prefall89**
~/.environment	**~/.environment.prefall89**
~/.path	**~/.path.prefall89**
~/.startup.X	**~/.startup.X.prefall89**
~/.startup.tty	**~/.startup.tty.prefall89**
~/.logout	**~/.logout.prefall89**

New standard Athena files have been placed into the user homes by copying template files from the system directory **/usr/prototype_user**:

~/.cshrc
~/.login

These user files mainly transfer control of the session to the following Athena session files in the system directory **/usr/athena/lib/init**:

cshrc (handles global environment setup and C shell setup)
xsession (handles sequence of activities in window-based session)
login (handles sequence of activities in tty-based session)

2.0.3 What should you do?

The file changes listed above provide a default session. If you have revised your configuration files in earlier releases and want to continue to use these tailorings in the new system, you should read the rest of this chapter, especially the Section 2.5 *Converting old configuration files to 6.3*.

In particular, **you should** *not* **simply rename the moved files to their old places -- the old ways of doing things do** *not* **work anymore**. For example, **~/.login.mine** is not supported -- the commands from this file should be disbursed into appropriate new files. Also, you should usually *not* change the system-supplied files (*.login, .cshrc*), even if you revise your own configuration files.

2.1 Overview of the new session structure

Note that there are now two distinct kinds of sessions:

- If you login via the new workstation interface (the **xlogin** interface described below), you are in a "window session" or "X session".

- If you login in any other way (e.g., via dialup, remote login, telnet), you are in a "tty session" or "login session"

Note that the **~/.login** file is only called in a "tty" session now. Nevertheless, the default "window session" closely resembles the Release 6.2 default behavior.

Managing the sessions behind the scenes are several system programs that you will usually not need to know much about. They are summarized here:

- For "window" sessions, the program **xlogin** handles the initial login window (explained below) and, once the user has passed the login window, manages the console window and calls **Xsession** to run the user session. The **Xsession** program runs an appropriate session file (usually the system file **/usr/athena/lib/init/xsession**) to control the session.
- For "tty" sessions, the program **/bin/login** handles the initial login sequence and, once the user has supplied a valid username and password, starts a special **csh** process that runs the user's **~/.cshrc** and **~/.login** files. When the user exits the session (by typing "logout"), **/bin/login** runs appropriate logout activities.

As in previous releases, the initial login program for "window" sessions is run within the **toehold** program, which also activates and deactivates the workstation, to keep the workstation screen from burning out and to keep the workstation from using power unnecessarily. In future releases, this may change, because **xlogin** is actually one component of a larger system called **xdm** ("X display manager") which is still unfinished but should supersede **toehold**.

2.2 The new "login" window

In Release 6.3B, the initial session window for "tty" (i.e., dial-up and remote login) sessions is the same as in earlier releases, just prompting for username and password.

The initial window for "window" sessions, however, has changed significantly. The interface (recognizable as the display with the Athena owl in it) used to be a 'login xterm' and is now a separate window. As in previous releases, to log in you just type your username and password with carriage returns. But now the login interface also includes two mouse-selected options:

- "Click here to register for an account" -- runs the account registration program
- "Click here for a non-standard session" -- presents a menu of options that allow users to run session files other than the default Athena file or their own default session file

The options for non-standard sessions are as follows:

- DEFAULT -- runs the standard Athena session (if a user **~/.xsession** file exists, **Xsession** uses this file as the session file; otherwise, **Xsession** uses the default system file **/usr/athena/lib/init/xsession**; this is the same as logging in without selecting the "non-standard" option at all)
- SYSTEM -- runs the standard Athena session, ignoring **all** user files (**Xsession** automatically uses the default system file **/usr/athena/lib/init/xsession**, and creates an environmental variable named $NOCALLS; with this variable set, the system file and all subsidiary system files ignore all user customization files; this option is especially useful for users who have made some incorrect customizations that prevent them from logging in -- the "SYSTEM" option assures a standard successful login)
- CUSTOM -- runs a session file that the user identifies (the file should be an *executable* shell script that has all the commands needed to run a successful session -- including starting a window manager if one is needed; a typical use of this option is to execute a small script, named **~/.fast** for example, which allows quick login for a specific task such as just reading mail)

In essence, each of these options identifies which session file **Xsession** runs. Note that any session file must have world execute permission and must be a valid shell script.

2.3 The "console" window

The **xlogin** program creates a special window named the *console window* to display status messages during a workstation-based session. The console window:

- is read-only (i.e., it cannot be written to -- hence no dialogue-based activity can be run through the console window)
- serves as both a "console window" and as the device /dev/ttyv0, so status ("blackbar") messages as well as any **talk** or **write** requests are sent to this window by default (once you *respond* to a **talk** request at an xterm window, future messages in the same conversation will be sent to that xterm window)
- is automatically placed in the upper-right-hand corner of the screen (to override this, move the window to a new location *twice*)
- can be iconified like any other X window
- has a button labeled HIDE which forces it to disappear completely from the screen (i.e., not simply iconified) until another message is sent to the window
- automatically appears (even if it has been hidden or iconified) whenever a new status message is sent to it
- can be *forced* by a user to reappear by issuing the **show_console** command at any xterm prompt
- cannot be gotten rid of completely

2.4 Logging out and the 'logout' button

As part of the standard "window" session, the system also runs a program called **xlogout**, which displays a small window button labeled "Logout" in the bottom-right-hand corner of the screen.

To end a window-oriented session (i.e., to "log out"), you can type **logout** at any xterm window, or you can click on the "Logout" button. If you click on the "Logout" button, the system pops up a window in the middle of the screen asking you whether you really want to log out (this prevents accidentally hitting the button from simply logging you out). If you click on the confirmation button, the system ends the session and logs you out.

A session is definitely ended when the following message appears in the console window:

```
Session terminated; bye ...
```

If you have logged out and this message has not appeared after a short while, something is wrong. (This usually occurs when **xlogin** cannot recover from some internal problem, and cannot complete the logout process. This generally requires rebooting the workstation to fix it, for normal public workstations.)

2.5 Converting old configuration files to 6.3B

This section describes how to update your old customizations to use the new configuration files.

Before Release 6.3B, the standard set of configuration files allowed you to customize your account with two files, **~/.cshrc.mine** and **~/.login.mine** (some users also customized **~/.cshrc** and **~/.login**). The new set of configuration files allows more flexibility and divides the functionality among a larger set of files.

Remember that the distinction now made is that a "window" session is a session begun on a workstation and involving multiple windows. A "tty" session is any other kind of session, including login via dialup, rlogin, rsh, or network (e.g., **telnet**).

With the new set of configuration files, **.cshrc.mine** is still used but, because of changes in the setup of the shell initialization files, some of your changes may need to be modified, or might work better in another file.

Here are some general guidelines for those wishing to convert their customizations to the Release 6.3B session model:

- **~/.login.mine obsolete.** The **~/.login.mine** file is no longer used; instead, its functions are taken up primarily by **~/.startup.X** or **~/.startup.tty**, depending on whether you want the commands to take effect in a "window" session or a "tty" session.
- **Setting your path.** If you want your default command search path (i.e., the $PATH environment variable) adjusted for your entire session, you can do this in a file named **~/.path**. This file should contain one command that sets the

shell variable $path (the command can use the shell variables $athena_path and $bindir). For example, you might use:

```
set path=( . ~/$bindir /usr/sipb/$bindir $athena_path)
```

Note that, if the **~/.path** file exists, the system uses its contents *instead* of setting the default Athena user path. Hence, you should *not* use a command like the following in the **~/.path** file, as the previous value of $path will not contain all the directories you should include in your path when **~/.path** is run:

```
set path=($path /usr/sipb/$bindir)
```

- **Setting environmental variables.** If you have other environment variables you wish to set (i.e., using **setenv**) at the beginning of any session ("window" or "tty" session), you can put commands into a file named **~/.environment**. For example:

```
setenv EDITOR emacsclient
setenv ROGUEOPTS "jump,passgo,name=Hubert,fruit=pineapple"
```

- **Setting shell variables/aliases.** You should put any changes to shell variables, as well as any aliases that you want to use in all of your shell windows, in your **~/.cshrc.mine** file; this is the only configuration file that the system will run for every new shell. For example:

```
set lineedit
```

- **Attaching file systems.** You can move your **attach** commands to either your **~/.environment** file, or to both **~/.startup.X** and **~/.startup.tty**. (If you put it in only **~/.startup.X**, for example, you will not automatically attach these file systems when you log in to the dialup servers; however, if you attach lockers in your **~/.environment** file, you can incorporate these files into your default path directly).

- **Starting up other windows.** This should be done from **~/.startup.X**; this is the main purpose of this file. A window manager (by default, **uwm**) is started up automatically, so you do not need to include this command. However, you should put other X startup commands such as **xterm** or **emacs** in this file. Remember to start all these commands in the background, or your session may hang (the system will wait for each process to end before going to the next one -- you will be able to use the windows that do start up, but you will probably have problems logging out).

- **.logout commands.** In the new session model, the commands in your **~/.logout** file are run when you log out of any kind of session ("window" or "tty"), so the commands in this file should be appropriate to either kind of session. Note especially that, for a "window" session, the logout sequence does not have an interactive window associated with it, so the logout commands cannot rely on such a window. For example, you should not put a **clear** command in your **~/.logout** file, or you will get warning messages (about a "non-zero exit status") every time you log out from a "window" session.

- **Avoiding exit.** You should not issue an **exit** command from any session configuration files that will be run in a "window" session, because this will cause the session to terminate.

- **Conditional statements.** To help you selectively issue commands depending on what kind of session you are in, you can include conditional statements using the $XSESSION system variable (this variable is set if you are in a "window" session, and is not set if you are in a "tty" session). For example:

```
if ($?XSESSION) setenv WINDOW_MANAGER mwm
```

Other new features you should know about are described in the document **"Dotfiles" : configuring your session in Release 6.3**. To view that document, use the **attach** command to attach the **athenadoc** locker, move to the directory **/mit/release/athenadoc**, and use the **viewdoc** program to examine the contents of the file **dotfiles.PS**:

```
athena% attach athenadoc
athena% cd /mit/athenadoc
athena% viewdoc dotfiles.PS &
```

Also, hard copies should be available in public clusters.

3. Changes of Interest to General Users

3.1 MWM: The Motif window manger

MWM WILL BE THE DEFAULT WINDOW MANAGER STARTING JANUARY, 1990.

3.1.1 Why MWM?

A window manager allows users and programmers to control placement, size, iconification, and other elements of windows. Athena's current default window manager is **uwm**. Recently, the X Consortium (which provides standards for X-window-based systems) wrote the ICCCM (InterClient Communications Conventions Manual). **Uwm** does not comply with the ICCCM. In order to be able to run applications from outside Athena and have our applications run out there, we must switch to a window manager that meets these conventions. The Open Software Foundation has provided us with **mwm**. **MWM WILL BE THE DEFAULT WINDOW MANAGER STARTING THIS JANUARY**. So you may want to start using it now.

3.1.2 How to get it & what it's like.

How can you start using **mwm** now? Simply put the following line in your *~/.environment* file:

```
setenv WINDOW_MANAGER mwm
```

The next time you login, your windows will look slicker.

To put it coarsely, mwm is closer to Macintosh behavior than uwm. The window have titlebars, buttons that produce pull-down menus, and graphic icons, among other features. Point your mouse at the window button in the upper left corner of an **mwm** window and hold down a mouse button and the resulting menu will provide you with most actions that you might be interested in performing to a window (circling up/down, resizing, iconifiying, moving, killing). Your mouse can also drag a window around your screen by its titlebar, and resize a window by grabbing and moving the appropriate side or corner of the border. Watch for a document on **mwm** in Athena's **olh**.

Similar to uwm, you can customize mwm using .Xresources and .mwmrc files. The Athena default mwm resource description is in */usr/lib/X11/system.mwmrc*. The **mwm** man page describes the extensive options.

3.2 OLH: Athena's new documentation delivery system

Athena's documentation will be changing from the current system of on-line *essential* and *more* documents to **OLH**. Now, to get documentation you just type **help**. OLH is menu driven and self-explanatory. An X version will be available during the fall. Not just Athena documentation staff writing will be accessible. The Athena consultants' amazing answers to often asked questions are available through OLH, and other kinds of useful information will be added to OLH in the coming months.

3.3 emacs: new options!

Emacs now supports all standard X toolkit switches. Especially significant is the disappearance of previously supported flags which are replaced with these new ones:

-r	**reverse video**	**is now -rv**
-b	**border width**	**is now -bw**

A full list of the new flags is on page 4 of the system release notes (/mit/release/6.3/notes). See Section 1.1 for how to access these notes.

3.4 Scribe updated to version 6, version 7 available

The command **scribe** now gets you Scribe version 6. Scribe version 5 has been moved offline to the locker *scribe5*. Scribe 6 has new attributes for multiple-column environments, can put footings below tables, and has new attributes, commands, and parameters for merging PostScript graphics into Scribe documents. For more information, check out the **Supplement for Version 6** in the Scribe section of the documentation racks in the public clusters.

Scribe version 7 is available in the locker *scribe7*. Scribe 7 has the nifty ability to perform vertical justification, so your very short documents can fill up single pages in a pleasing manner. Scribe 7 can also put landscape (sideways) pages in portrait (normal) documents and has many enhancements to tables, figures and graphics. For more information, check out the **Supplement for Version 7** in the Scribe section of the documentation racks in the public clusters.

3.5 xmh: some changes

Xmh has had several bug fixes and is now more user-friendly. Errors, such as exceeding quota while trying to incorporate mail, pop up a dialog box. For users who have customized xmh resources: if any of your resource specifications use the Command class, the new xmh implementation has changed some widget classes from Command to Toggle. See the page 16 of the system release notes for more details. Section 1.1 of this document explains how to get to these notes.

3.6 xquota

You can now check your file storage usage with the command **xquota**. The window that it uses describes graphically and numerically the amount of storage you have used. There is a help button below the window. Clicking the left mouse button on the window pops up a second window with more extensive information.

4. Changes of Interest to Programmers

4.1 Saber-C 2.1!

Typing *saber* on Vax or RT now gets you version 2.1 of the Saber-C interpreter and debugger. Among other improvements, you no longer need to enter commands in syntactically correct C, making it much smoother to use. Typing *xsaber* on a Vax workstation will get you an X window interface to Saber 2.1.

4.2 AFS: the Andrew File System

The Andrew File System (or AFS) is now available. People who manage lockers to which many people need read/write access should consider using AFS. It allows better control over access lists and quota and easy facility for providing back-ups. Users will find it faster and more immune to temporary network problems. Faculty and developers who are interested should contact the faculty liaisons Anne LaVin (*lavin*) or Naomi Schmidt (*nschmidt*) at Project Athena.

4.3 X10 moved.

The X10 binaries have been moved into the *x10* locker. *xswitch* will continue to works as normal, but in the case that the locker is not attached, it will first attempt to attach it. If you are on an RT, it will inform you that the server will only work on a Vax.

4.4 New version of *RCS*

A new version of RCS (Revision Control System), from Berkeley's 4.3BSD-Tahoe release, has been installed. All Athena changes have been added. Added 'Header' capabilities are of note. See the system release notes for more detail.

4.5 New version of *make*

Also from Berkeley's Tahoe release comes a new version of *make*, which uses user's environment variables as defaults and has a new variable 'MACHINE'. See the system release notes for more detail.