

Your Andrew Unix Account

This document contains the following sections:

- [Introduction](#)
- [Your Andrew Quota](#)
- [The Plan File](#)
- [.login & .logout Files](#)
- [Your Backup Directory](#)
- [Prototype Files & Directories](#)
- [Search Paths](#)

For information related to this topic refer to:

- [Andrew Files and Directories](#)
(<http://www.cmu.edu/computing/doc/unix/afs-files.html>)
- [Andrew Linux Support Statement](#)
(<http://www.cmu.edu/computing/doc/os/linux.html>)
- [Andrew Solaris Support Statement](#)
(<http://www.cmu.edu/computing/doc/os/solaris.html>)
- [Increasing and Managing Your AFS Quota](#)
(<http://www.cmu.edu/computing/doc/accounts/quota/index.html>)
- [Logging into and out of AFS](#)
(<http://www.cmu.edu/computing/doc/unix/afs-login.html>)
- [Project Volumes](#)
(<http://www.cmu.edu/computing/doc/accounts/project-volume/index.html>)
- [Setting Directory Protections and Using PTS Groups in Andrew](#)
(<http://www.cmu.edu/computing/doc/unix/pts-groups/index.html>)
- [Using Unix and the Andrew System](#)
(<http://www.cmu.edu/computing/doc/unix/unix-andrew/index.html>)

Last Updated: 11/7/05

Introduction

This document describes your Andrew account and your Andrew Unix account. By using the information in this document you can customize and manage your account so that it behaves the way you want each time you login to a unix server or workstation. For more information about using Andrew and UNIX see *Using UNIX and the Andrew System* (<http://www.cmu.edu/computing/doc/unix/unix-andrew/index.html>).

Last Updated: 11/7/05

Your Andrew Quota

What is Your Quota?

Your quota, or file space allocation, is the amount of storage space allocated to you in AFS, the file system used for your Andrew account. The amount of storage space available in AFS is very large, but it is not infinite. To make sure that everyone gets a fair share of the space available, a limit is placed on the space given to each user. Students, faculty and staff members are assigned an initial quota of 500 MB (megabytes). If you need more space, you can increase your quota to up to 1 GB (gigabyte) using the *Quota Increase Tool* (<http://www.cmu.edu/computing/doc/accounts/quota/index.html>).

The Quota Increase Tool

The *Quota Increase Tool* (<http://www.cmu.edu/computing/doc/accounts/quota/index.html>) (<http://www.cmu.edu/computing/doc/accounts/quota/index.html>) is a web interface available from the My Accounts tab of the *Carnegie Mellon Web Portal* (<https://my.cmu.edu/>). The tool allows you to:

- Check your current AFS quota
- View the percentage of your quota in use
- View the maximum quota allowed for your account type (in megabytes)
- Increase your AFS quota

Note: Users working on specific projects are often able to obtain a project directory (or "project volume"). Project volumes are areas of disk space which are separate from the user's private disk space. For more information on obtaining a project volume, read *Project Volumes on Andrew* (<http://www.cmu.edu/computing/doc/accounts/project-volume/index.html>).

What to do if your quota is full

If you have reached or are in danger of reaching your maximum quota, review some tips for *Managing your Quota* (<http://www.cmu.edu/computing/doc/accounts/quota/index.html>).

Last Updated: 11/7/05

The Plan File

The plan file helps you to control the information that others see when they use the finger command to lookup your account.

By default, the following information is included for student and faculty/staff accounts:

Student Account	Faculty/Staff Account:
name userIDs (at Carnegie Mellon) class department affiliation login name email address	name userIDs (at Carnegie Mellon) department affiliation login name email address office phone and address

- Use the University Directory to set preferred information for name or email address. The directory privacy settings can also be used to show additional information such as home address and phone.
- Use the "cyrfinger on" and "cyrfinger off" commands to toggle the ability to have finger show whether you have new email.
- Use the plan file to store any other information that you would like others to see when they use the finger command on your account.

For more information about creating or updating your plan file, read [Creating an Andrew Plan File \(http://www.cmu.edu/computing/doc/contributed/planfile.pdf\)](http://www.cmu.edu/computing/doc/contributed/planfile.pdf).

Last Updated: 11/7/05

.login and .logout Files

What .login and .logout Do

Your **.login** and **.logout** files send commands to the UNIX shell, often setting or unsetting UNIX environment variables that control how certain UNIX commands behave.

How .login and .logout Work

When you log in, the UNIX shell consults your **.login** file and runs the commands in it. In particular, the UNIX shell reads your **.login** file to find other UNIX commands and to determine what type of windowing system you want to run.

Similarly, when you log out, the UNIX shell reads your **.logout** file and runs the commands in it.

Global .login and .logout Files

Sourcing the global **.login** and **.logout**

Your **.login** and **.logout** files probably contain lines which "source," or run, the commands in the global **.login** and **.logout** files. These lines are of the form:

```
source /usr/local/lib/global.login and source /usr/local/lib/global.logout
```

Where to put the source commands

The source line for **global.login** should be the first command in your **.login** file. If you make changes to **.login**, make them below this line.

The source line for **global.logout** should be the first command in your **.logout** file. If you make changes to **.logout**, make them below this line.

The global.login file

The **/usr/local/lib/global.login** file sets these things:

- The search path your UNIX shell will use to find programs.
- The editor UNIX uses by default.
- Your time zone variable.
- A tmp file used for zephyr communications.

The global.logout file

The **global.logout** file executes the **unlog** command, telling the AFS distributed file system to delete your authentication tokens from this particular session. The **global.logout** file also stops **zephyr** and cleans up any local *Xauth* cookies if appropriate.

Making Changes to Your .login

Setting your terminal type

If you want AFS to ask about setting the terminal type when you log in via telnet, add the following code to the end of your **.login** file:

```
if ($term == "" || $term == "network" || $term == "telnet" || $term == "su")
then echo -n "Terminal type [vt100] ? " set term=$< if ($term == "") set
term="vt100" set -I Q endif
```

Note: You may set the terminal type to something other than vt100 (e.g., zenith, vt220, etc.).

Setting the windowing environment

All new Andrew accounts automatically start the X11 windowing system and fvwm2 whenever the user logs in. This is accomplished by means of the line:

```
source /usr/local/lib/global.startwindow
```

in the user's .login file.

There are, however, alternate window managers a user may choose to run, twm (Tab Window Manager), or a user may decide not to use any window manager and just work from the system console. Which windowing environment you choose will depend on your personal preference, as well as the capabilities of the machine you may be using.

Computing Services recommends using X11 and fvwm2. All the Andrew workstations in public clusters work very well with X11 and fvwm2.

For more information about the default Andrew window environments, see [The Andrew Window Environment \(http://www.cmu.edu/computing/doc/unix/x11/index.html\)](http://www.cmu.edu/computing/doc/unix/x11/index.html).

Specifying command line arguments to the Xserver

Your .login file is also the place where you issue commands to the Xserver upon login, using the environment variable XSERVERARGS.

For example, if you want to make sure that a GrayScale server is started when you login to the machine named freehold, which you know has a GrayScale display, you could put something like the following in your .login file:

```
#if on freehold, make sure I get a GrayScal # server if (`hostname` ==
freehold.andrew.cmu.edu) then setenv XSERVERARGS "-cc 1" endif
```

The -cc switch would be used when running the MIT X servers (like SPARCstations and other non-DEC servers), while the -class switch would be used when running the DEC X servers.

The Display classes are as follows:

- #define StaticGray 0
- #define GrayScale 1
- #define Static Color 2
- #define PseudoColor 3
- #define TrueColor 4
- #define Direct Color 5}

Note: These arguments for the Xserver must appear before the source `/usr/local/lib/global.startwindows` line, which should appear at the end of your .login file.

Modifying your search path

You can modify the search path that UNIX uses to find programs you want to run. See the *Search Paths: Using Search Paths on Andrew* section for an explanation of the search path and how to modify it in your .login file.

Logging Out vs. Quitting

The difference between logging out and quitting is that logging out takes you completely out of the AFS system, while quitting takes you out of the window manager and back to the system prompt.

Some users may prefer to just quit the window manager without logging out of the system (this is useful if you want to do things outside of X, like restart X to test a change you made without having to log back in).

In order to quit, rather than log out:

1. Add the following line to your .login file, before the line that sources the global.startwindows:
setenv NO_LOGOUT_ON_QUIT ""

Note: This environment variable will have an effect only if you have started your windowing environment via the global.startwindows mechanism, and not until your next login. Once you have set this variable, choosing Logout/Quit from the menu will quit the windowing environment and place you back at the system prompt. From there you will have to **type logout to log out of the system.**

Last Updated: 11/7/05

OldFiles: Your Own Backup Directory

What is OldFiles?

OldFiles is a subdirectory in your home directory that contains a complete copy of the full directory tree (that is, all the directories, subdirectories, and files) in your home directory as it appeared sometime within the previous 24 hours. OldFiles is read-only, so it does not count in your quota.

If you want to recover a file that you accidentally deleted or changed within the past day, you may be able to retrieve the older copy from the OldFiles directory. The OldFiles directory has the same protections as your home directory.

Note: If more than 24 hours pass before you notice that you deleted the file, a copy of the file will no longer be in OldFiles.

A new copy of OldFiles is made every night approximately between 6 P.M. and 4 A.M. See the backups help file for information on retrieving deleted files that are not in your OldFiles directory.

Using OldFiles

OldFiles is like any directory in that you can copy and list files in it, change to it with the `cd` command, and so forth. However, OldFiles is read-only, so you cannot make changes to any of its contents. This means you cannot use the move (`mv`) command to place files in your home directory--you must use the copy (`cp`) command instead.

To locate a file in your OldFiles directory,:

1. From your home directory, type `cd OldFiles` and press Return to change to the OldFiles directory.
2. Type `ls` and press return to see a listing of the files in OldFiles (or type `ls -l` to see a long listing of the files, including their dates).
3. Once you have located the file you want, use the `cp` command to copy back into your home directory.

Example 1

If you accidentally deleted a file called "notes" and you want to copy it from the top level of the OldFiles directory into your home directory, type:

```
cp OldFiles/notes ~
```

and press Return.

Example 2

To rename the file "old-notes" while copying it to your home directory, type:

```
cp ~/OldFiles/notes ~/old-notes
```

and press Return.

Last Updated: 11/7/05

Prototype Files and Directories

What are prototypes?

Prototypes are examples of files and directories as they appear in a new Andrew account. If you think something is wrong with your Mailbox, PrintDir, private, or public directories, or your preferences, .login, .logout, or .cshrc files, you can copy a prototype into your own account to replace the incorrect file or directory.

Using prototypes

The `/afs/andrew/data/db/account` directory contains all of the files used to create a new Andrew account. The files in this directory are:

- .login
- .logout
- .cshrc
- preferences
- bin
- private
- public

Because several of these files contain a period before the file name (often referred to as "dot" or "hidden" files), you will need to use the `ls -a` command to list all of the files in this directory. (This command lists both visible and hidden files.)

To copy a file from the `/afs/andrew/data/db/account` directory into your home directory,:

1. Type `cp /afs/andrew/data/db/account/<filename>` at the system prompt and press **Return**.
 <filename> is the name of the file you wish to copy.

To avoid overwriting something you may want to keep, it is a good idea to rename the file you want to replace before copying the new one. For example, if you want to copy the proto version of .login, you might change the name of your existing .login file to .login.old. That way, you will be able to retrieve information from your old file that you want to include in the new one. See the [Andrew Files and Directories \(http://www.cmu.edu/computing/doc/unix/afs-files.html\)](http://www.cmu.edu/computing/doc/unix/afs-files.html) document to learn how to rename a file.

Where to Copy Prototypes

Be sure to copy prototypes into your home directory. They won't work if you copy them anywhere else.

Last Updated: 11/7/05

Search Paths: Using Search Paths on Andrew

What is a Search Path?

A search path is the list of directories in which the UNIX shell looks to find programs that you want to use. When you issue a command, the UNIX shell looks in each directory listed in your search path in sequence and runs the first program that matches the command you gave. A global search path is set for you automatically when you login and you can add directories to it if you wish.

Global Search Path

The global search path is the search path which Andrew System administrators provide for all Andrew users. In general, it includes these directories:

on Andrew Linux:

```
/usr/local/bin
/usr/edubin
/bin
/usr/bin
/usr/contributed/bin
```

on Andrew Solaris:

```
/usr/local/bin
/usr/ccs/bin
/opt/SUNWspro/bin
/usr/java/bin
/usr/edu/bin
/usr/ucb
/bin
/usr/contributed/bin
/usr/openwin/bin
```

A bin directory is where binary, or compiled, versions of programs are conventionally stored. Your own bin directory is always included in the global search path, so you never need to add it to your personal search path. The global search path is maintained in the file **/usr/local/lib/global.path**, which is referenced, or sourced, by the file **/usr/local/lib/global . login**.

Each time you log into Andrew, the commands in the .login file are sourced. If you examine the copy of .login in your home directory, you will see that the first line in the file is:

```
source /usr/local/lib/global.login
```

his source line, which includes standard configuration information, ensures that all currently supported system directories are available. Changes to your .login file should be made after this line.

Adding Directories to the Search Path

To add a directory to your search path:

1. Open your `.login` file for editing.
2. After the line, **source /usr/local/lib/global.login** , add the new search path. Search path lines are of the form:

set path=(\$path <path>)

Note: In the set path line above, `$path` refers to the path already set by the `global.login` file. This line tells the computer, "Set my path to be: the path set by the `global.login` file `$path` plus `<path>`, in that order."

3. Save your changes.

Example

To add a directory called `~hb0v/bin` to your search path, you would need to add the line:

```
set path=($path ~hb0v/bin)
to your .login file somewhere after the line:
source /usr/local/lib/global.login
```

Controlling the Order of the Search Path

The order in which the search paths appear in your `.login` determines the order in which the directories are searched. The normal method for setting the search path searches the system directories first and then searches the path you specify because it is the last directory in your search path line. However, there may be times when you want to search a specific directory before searching the system directories. To set your path to search a specific directory before the system directories:

1. Use a search path of the form:
Set path= (<path> \$path)

Note: In the set path line above, `$path` refers to the path already set by the `global.login` file. This line tells the computer, "Set my path to be: `<path>` plus the path set by the `global.login` file, in that order."

Example

One of your professors may want your class to use a modified version of a program that is already installed on one of the default paths. In this situation, you will need the UNIX shell to find the class's version of the program before the system version. If your class directory is `15-211` and the directory where the class version of the program is stored is `/afs/andrew/cs/15-211/bin` , you could modify your search path by adding the following line to your `.login` file somewhere after the `source /usr/local/lib/global.login` line:

```
set path=(/afs/andrew/cs/15-211/bin $path)
```

Everything about this line is the same as the other example except that a directory has been added at the beginning of the system path.

Last Updated: 11/7/05