

# **UW Computer Science and Engineering**

---

## **AFS Beginner's Guide**

---

May 1996 - Warren Jessop (revised January 2001)

---

# READ READ READ... DEMISE OF AFS FILE SERVICE

We interrupt this AFS web page for a special bulletin....

## Latest News

For the latest news on the AFS to NFS migration and the demise of AFS, see Support FAQ #19: [rtfm.cs.washington.edu/cgi-bin/wreq/req2?showfaq-faq-1-19](http://rtfm.cs.washington.edu/cgi-bin/wreq/req2?showfaq-faq-1-19)

## Executive Summary

Two new Linux (UNIX/NFS) research servers, administered by the systems group, have been installed to *replace* the current AFS servers and to serve files that are currently being served by AFS. Directory names will correspond as follows:

Old AFS Directory	New NFS Directory
-----	-----
/afs/cs/homes	/homes/sys
/afs/cs/projects	/projects/sys
/afs/cs/sources	/sources/sys

On Sunday, February 4, 2001, all AFS volumes (Section 1.6 [Servers], page 9) will be copied from the AFS servers to the NFS servers by the support group, as detailed below. Naturally, if the AFS volumes are subsequently updated with new/edited files after the initial copy, they may need to be copied again. We (support) will be glad to provide procedures for doing copies of AFS volumes.

On Sunday February 18, 2001 the LAST BACKUP of the departmental AFS filesystems will be taken; if you have an AFS account you may continue to use those servers, but after that date anything you store there may be lost permanently if there is some type of system failure.

On Sunday April 1, 2001 the AFS servers will be powered down for good. After that date restores from AFS backup tapes will not be possible.

AFS provides much richer file access control mechanisms than UNIX (Chapter 2 [Access-Rights], page 11), hence there cannot be a perfect mapping from permissions in an AFS directory to those in its UNIX copy. The support group will make an initial, but necessarily *imperfect*, effort to map AFS permissions to the UNIX world when AFS volumes are copied to UNIX directories the first time. Subsequent fine tuning, however, will be up to the owners of the data. Migration details are given below.

If you currently log into an AFS client and your home directory is on an AFS server, you will need to take action as described below under Details.

AFS "mount points" (Section 3.2 [Backup Volumes], page 13) will be handled by replacing them with symbolic links on the NFS servers. The exact method of dealing with AFS mount points is given below.

## Details

### Home directories

If the the home directory you have when you log into a UNIX system is on an AFS server, you will at some point need to change your home directory to one on an NFS server. Everyone already has an NFS home directory. You just need to send a message to support@cs to effect the change.

The sooner you do this, the better. Obviously it makes no sense to wait until after the AFS servers have been shut down; on the other hand, experience tells us that some users will do exactly that.

### Copying files to new NFS servers

All afs volumes will be copied to the new NFS servers as follows:

- All home directories that appear under `‘/afs/cs/homes’` are actually AFS mount points that point to home directory volumes. Each home directory volume (e.g. jouser’s volume, called `user.jouser`) will be copied and made available on all research UNIX systems under `‘/homes/sys’` and on NT/W2000 systems under `‘\\ntdfs\cs\unix\homes\sys’` (e.g. `‘/homes/sys/jouser’` and `‘\\ntdfs\cs\unix\homes\sys\jouser’`). The owner and group of `‘/homes/sys/jouser’` will be set appropriately; mode will be read/write by owner only.
- The UNIX owner and group of all the non-home (`‘/projects/sys’` and `‘/sources/sys’`) directories will be set to something “reasonable.” If you have special requests for particular settings of owner and group on particular volumes, or for new UNIX groups to be created to mimic AFS groups, please let us know.
- A number of AFS project volumes projects appear under `‘/afs/cs/projects’` as AFS mount points (e.g. `proj.spin`). These include:

directory (mount pt.)	AFS volume
-----	-----
contrib	proj.contrib
course	proj.course
dyncomp	proj.dyncomp
kimera	proj.kimera
lis	proj.lis
luxor	proj.luxor
metacrawler	proj.metacrawler
mojava	proj.mojava
opal	proj.opal
porc	proj.porc
porcupine	proj.porc
prism	proj.prism
sio	proj.sio
spin	proj.spin

visual	proj.visual
webos	proj.webos
syn/amit	proj.syn.amit
syn/cardwell	proj.syn.cardwell
syn/savage	proj.syn.savage
trace/bchen	proj.trace.bchen
trace/jlo	proj.trace.jlo
trace/romer	proj.trace.romer

Each of these will appear under `‘/projects/sys’` (`“\\ntdfs\cs\unix\projects\sys”`) with the directory names listed above (e.g. `‘/projects/sys/spin’` and `‘/projects/sys/trace/jlo’`).

- Other AFS project volumes will be copied to one of three directories, `‘/projects/sys/.proj[123]’` (e.g. volume `proj.spin.www` will be copied to `‘/projects/sys/.proj1/spin.www’`), as shown here:

AFS volume	copied to
proj.meta.c	/projects/sys/.proj1/meta.c
proj.meta.d	/projects/sys/.proj1/meta.d
proj.meta.f	/projects/sys/.proj1/meta.f
proj.meta.g	/projects/sys/.proj1/meta.g
proj.meta.h	/projects/sys/.proj1/meta.h
proj.meta.i	/projects/sys/.proj1/meta.i
proj.opal.chase	/projects/sys/.proj1/opal.chase
proj.opal.jamrozik	/projects/sys/.proj1/opal.jamrozik
proj.opal.nara	/projects/sys/.proj1/opal.nara
proj.spin.hoffman	/projects/sys/.proj1/spin.hoffman
proj.spin.m3	/projects/sys/.proj1/spin.m3
proj.spin.mef	/projects/sys/.proj1/spin.mef
proj.spin.merge	/projects/sys/.proj1/spin.merge
proj.spin.myrinet	/projects/sys/.proj1/spin.myrinet
proj.spin.owa	/projects/sys/.proj1/spin.owa
proj.spin.test	/projects/sys/.proj1/spin.test
proj.spin.www	/projects/sys/.proj1/spin.www
proj.opal.tiary	/projects/sys/.proj2/opal.tiary
proj.opal.voelker	/projects/sys/.proj2/opal.voelker
proj.spin.archive	/projects/sys/.proj2/spin.archive
proj.spin.artjg	/projects/sys/.proj2/spin.artjg
proj.spin.becker	/projects/sys/.proj2/spin.becker
proj.spin.bershad	/projects/sys/.proj2/spin.bershad
proj.spin.bin	/projects/sys/.proj2/spin.bin
proj.spin.build	/projects/sys/.proj2/spin.build
proj.spin.chase	/projects/sys/.proj2/spin.chase
proj.spin.cvsroot	/projects/sys/.proj2/spin.cvsroot
proj.spin.ddion	/projects/sys/.proj2/spin.ddion
proj.spin.dist	/projects/sys/.proj2/spin.dist
proj.spin.eaberg	/projects/sys/.proj2/spin.eaberg

```

proj.spin.pardy          /projects/sys/.proj2/spin.pardy

proj.meta.e             /projects/sys/.proj3/meta.e
proj.porc.manu          /projects/sys/.proj3/porc.manu
proj.porc.maureen       /projects/sys/.proj3/porc.maureen
proj.porc.nspring       /projects/sys/.proj3/porc.nspring
proj.spin.egs           /projects/sys/.proj3/spin.egs
proj.spin.egs.jdk       /projects/sys/.proj3/spin.egs.jdk
proj.spin.ericc         /projects/sys/.proj3/spin.ericc
proj.spin.fgray         /projects/sys/.proj3/spin.fgray
proj.spin.imenn         /projects/sys/.proj3/spin.imenn
proj.spin.matthai       /projects/sys/.proj3/spin.matthai
proj.spin.oyster        /projects/sys/.proj3/spin.oyster
proj.spin.rgrimm        /projects/sys/.proj3/spin.rgrimm
proj.spin.robs          /projects/sys/.proj3/spin.robs
proj.spin.savage        /projects/sys/.proj3/spin.savage
proj.spin.tian          /projects/sys/.proj3/spin.tian
proj.spin.whsieh        /projects/sys/.proj3/spin.whsieh
proj.spin.yasushi       /projects/sys/.proj3/spin.yasushi

```

- AFS source volumes (e.g. named `src.yyy`) will be copied to `‘/sources/sys’` (e.g. as `‘/sources/sys/yyy’`).
- Other AFS volumes names begin with `local`, `root` and `system`; these will be copied to one of three directories, `‘/projects/sys/.afs/local’`, `‘/projects/sys/.afs/root’`, or `‘/projects/sys/.afs/system’`.

## Handling AFS mount points

If an AFS mount point is encountered when copying directory hierarchies from AFS servers to NFS servers, it will be handled as described in this example:

Let the directory name of the mount point be `‘xxx’` and the AFS volume it points to be called `proj.xxx`. The directory (mount point) that appears in AFS will be replaced with a symlink on the destination NFS filesystem. The name of the symlink will be `‘xxx-DEAD-AFS-MTPT’` and its contents will be `‘#proj.xxx’`.

For example, `ls -ld xxx; fs lsmount xxx` on AFS would put out something like this (see `man fs_lsmount` on an AFS client for more info on listing mount points):

```

drwxrwxrwx 4 root spin 2048 Oct 30 1998 xxx/
'xxx' is a mount point for volume '#proj.xxx'

```

while `ls -l xxx` on NFS would put out something like this:

```

lrwxrwxrwx 1 jouser grad_cs 13 Jan 3 12:22 xxx-DEAD-AFS-MTPT -> #proj.xxx

```

It is up to the owner of each directory to create a correct link (and delete the “DEAD” link if desired). E.g. if the root of volume `proj.xxx` winds up in `‘/projects/sys/.proj1/xxx’` on the NFS server,

```
ln -s /projects/sys/.proj1/xxx
```

would recreate the desired link.

We now return you to your regularly scheduled web page...

## For the Information-Overloaded...

If you don't have time to read the first two parts of this guide (Chapter 1 [Intro], page 7 and Chapter 2 [AccessRights], page 11), at least read this first page. This is what you need to know if you have an AFS home directory or if you must access some other protected AFS directory. AFS is a type of file system enabled on most UNIX workstations in the department. You can work with AFS files the same as you would with "normal" UNIX files, using the same editors, compilers, and other tools. However, there are some significant differences you need to be aware of.

- All AFS files in the department have full path names that begin with `/afs/cs/`, e.g. joeuser's AFS home directory is `/afs/cs/homes/joeuser`.
- If you have an AFS home directory created since May, 1996, the files in it are readable only by you. A `public` directory has been created inside your home directory; if you put files in there they will be readable by everyone (Section 3.3 [HomeDir], page 14).
- If you have an AFS home directory created before May, 1996, you should check its protections using the `fs listacl directory` command (Section 2.4 [ExamineACL], page 12).
- You have to "log in" to AFS in order to access your files. On most workstations the login process is combined with the normal UNIX login, so all you need do is supply your AFS password; on a few you may need to use an AFS command to enable access to your files (Section 1.7 [Auth], page 10).
- You have at least two passwords: your normal good old UNIX password(s), and an AFS password, which can (and normally should) be the same as at least one of the UNIX passwords. When you got your UNIX account(s) you were given information on how to change your password(s); similarly, when you were given an AFS home directory you received information on changing your initial AFS password (Section 1.7 [Auth], page 10). The procedure for changing your AFS password is different from that for changing your UNIX password(s).
- Your environment may not be set right if you are not using the departmental standard `.cshrc` file. You may need to make sure `PATH` and `MANPATH` have been set correctly (Section 1.3 [UNIX Env], page 7).
- Sometimes you may need to access AFS files from a computer that does not have full AFS capabilities, via a so-call "translator" host. You need to know how to tell whether this is the case, and if so how to deal with this situation (Section 3.8 [AFSNFS], page 19).
- If you use `rsh`, `rcp`, or `rlogin` (or a command that uses any of them, such as `xrsh`), you may need to know that there are AFS versions of these commands (Section 3.9 [Rcommands], page 21).

## AFS Driver License Quiz

The following questions assume that your user name is 'joeuser' and that your home directory, called '/afs/cs/homes/joeuser', is kept on an AFS file server. Answers are in Appendix A [Answers], page 23

1. You've used the `ls` command to look at the characteristics of file '.secret' in your home directory:

```
% ls -l /afs/cs/homes/joeuser/.secret
-rw----- 1 joeuser joegroup 4698 May 1 11:25 .secret
```

True or false: no one except you and the system administrator can read this file.

2. You are editing a file in your home directory and suddenly you cannot seem to save your changes. What's happened?
3. The system says you have exceeded a space quota. How do you find out what your home directory space quota is?
4. You have a number of windows displayed on your workstation. Your friend Mikey drops by and wants to show you one of his files, which is only readable by him. He uses one of your command windows to get an AFS "token," as follows:

```
% klog mikey
Password: mikey's_AFS_password
% view /afs/cs/homes/mikey/coolfile
```

You notice after he leaves that processes that had been working in your other windows are getting messages like

The file access permissions do not allow the specified action.

How can it be that you are denied access to your own files?

# 1 Introduction

AFS (the AFS File System, a product of Transarc Corporation) is a distributed file system that operates on a worldwide network of client hosts and server hosts. An AFS client host can access files residing on an AFS server transparently, as if they resided on a local disk. This is similar to how NFS is set up locally in the CSE department. However, unlike NFS, AFS mandates the use of a truly global hierarchical naming scheme in which a file has the same path name on any AFS client anywhere in the world.

This document briefly presents information you will need to get started with AFS and notes some of the principal differences between AFS and the usual UNIX file systems such as NFS. A familiarity with UNIX filesystems and file structure is assumed.

## 1.1 Getting an AFS Account and Home Directory

You can read publicly accessible AFS files even if you do not have an AFS account, so not everyone will need an account.

If you already have a CSE research account you may also get an AFS account and directory. This will be referred to as your AFS *home* directory, although it need not be your *login* directory. Send mail to `afs@cs` to request an account. Your home directory will be assigned a space quota, usually 20 megabytes or less.

## 1.2 Getting an AFS Login Directory

Until AFS becomes ubiquitous throughout the department, your AFS home directory will not be your login directory for every host in the department. If you want your login directory to be your AFS home directory for a particular group of AFS client hosts, send mail to `afs@cs`. If your AFS home and login directories are one and the same, your quota can be set according to the kind of account you have or the project you are working on.

## 1.3 UNIX Environment and AFS Availability

In order to access AFS commands you should make sure your PATH environment variable has a `/usr/afsws/bin` component. Likewise, to access man pages, MANPATH should have a `/usr/afsws/man` component. If you are using the current departmental standard `.cshrc` file<sup>1</sup>, PATH and MANPATH will have these components; see Section 3.9 [Rcommands], page 21 for more information.

If `/usr/afsws` does not exist on the host you are logged into (e.g. `'cd /usr/afsws'` fails), the host does not have the ability to be an AFS client, or for some reason it has not been configured as an AFS client

Of the current CSE research hosts, all are or will eventually be configured as AFS clients except: DECstations running Ultrix, some Alphas, and some of the older Suns.

---

<sup>1</sup> This is the `.cshrc` that new home directories get a copy of. If your home directory was set up some time ago, you may want to compare your `.cshrc` to the copy of the standard file kept in `/cse/lab/dotfiles/dept.cshrc` on all research hosts

## 1.4 Cells and File Names

The root of the AFS file tree is `/afs`. The second component of the path is a *cell* name—usually a DNS dotted name. Thus,

```
/afs/cs.washington.edu
```

is the prefix to every fully qualified AFS path name in our local departmental cell. A cell is an administrative entity; below the second level each cell maintains its own organization.

Locally, a symbolic link under `/afs`, `cs -> cs.washington.edu`, allows path names to be abbreviated, e.g.

```
/afs/cs/homes/joeuser
```

can be used for the canonical name,

```
/afs/cs.washington.edu/homes/joeuser.
```

The following directories are below `/afs/cs`:

<b>homes</b>	home directories, e.g. <code>/afs/cs/homes/joeuser</code>
<b>projects</b>	project directories, e.g. <code>/afs/cs/projects/lis/ptolemy</code>
<b>sources</b>	source directories, e.g. <code>/afs/cs/sources/spec95</code>
<b>local</b>	locally added supported software, similar to <code>/usr/local</code>
<b>uns</b>	files contributed by students, faculty and staff, but not supported by lab staff.
<b>admin</b>	local AFS administration files
<b>system</b>	AFS system files: executables, libraries, man pages, etc.
<b>cse</b>	departmental directories, used by office and CS lab staff.

From an AFS client's point of view, cells other than the local cell (whose name is specified in `/usr/vice/etc/ThisCell`) are called foreign cells. A list of all foreign cells that a client can access can be found in `/usr/vice/etc/CellServDB`.

## 1.5 AFS On-Line Help and Man Pages

Some AFS commands are organized into “suites”, for example the **fs** suite. The first parameter of such a command is an *opcode* for the suite. All suite commands have an opcode `help`, which will list all the other opcodes, e.g. for **fs**:

```
% fs help | head -10           (only show the first few)
fs: Commands are:
apropos          search by help text
checkservers     check local cell's servers
checkvolumes     check volumeID/name mappings
cleanacl         clean up access control list
copyacl          copy access control list
debug            set debugging info
diskfree         show server disk space usage
examine          display volume status
exportafs        enable/disable translators to AFS
```

An opcode can be abbreviated by entering a prefix string long enough to make it unique, e.g. ‘`fs exa .`’ is the same as ‘`fs examine .`’. In addition, opcode aliases may be accepted. Adding an *opcode* after ‘`help`’ causes the aliases and command usage to be displayed for that opcode, e.g.:

```
% fs help examine
fs examine: display volume status
aliases: listvol lv
Usage: fs examine [-path <dir/file path>+] [-help ]
```

Most commands, whether part of a suite or not, also have a ‘`-help`’ parameter that will display usage.

In addition to the built-in command help, a set of man pages for AFS commands is kept in ‘`/usr/afsws/man`’. The man pages are organized in an unobvious way for the command suites. For example, instead of ‘`man fs`’ yielding a complete man page for `fs` and all its opcodes, it yields only an introduction to the `fs` suite. In order to know in detail what a particular opcode of `fs` does, you need to enter ‘`man fs_opcode`’ (note the ‘`_`’), e.g. ‘`man fs_examine`’. Note that the full opcode must be specified, and no aliases are allowed, e.g. ‘`man fs_exa`’ or ‘`man fs_listvol`’ will not work. To see a “whatis” list of all AFS man pages, enter ‘`man -k AFS:`’.

## 1.6 Servers and Volumes

The unit of storage on an AFS server is called a *volume*—a named subsection of a disk partition. Volumes consist of natural “chunks” of data; for example, each home directory is contained in a single volume. In using AFS you will almost never need to know which servers<sup>2</sup> contain particular volumes, and you will rarely need to be aware of the names of volumes.

A space quota may be assigned to an AFS volume. At this time, home directories have been assigned a range of quotas from 20MB to unlimited. To find the volume name, quota, and space usage corresponding to any AFS path, use the ‘`fs listquota path`’ command, e.g.

<sup>2</sup> There are 5 servers in the CSE department: 3 Alphas running OSF/1 3.2, and two IBM PowerPCs running AIX 3.2.5

```
% fs listquota /afs/cs/homes/joeuser
Volume Name      Quota   Used    % Used  Partition
user.joeuser     100000  91181   91%<<   78%    <<WARNING
```

To have a better “feel” for what AFS does, it may help to know that in addition to containing AFS volumes, the servers also maintain several databases:

- A *volume location database* that maps volume names to servers and disk partitions.
- An *authentication database* that contains usernames, encrypted passwords, last password change date, and password expiration date.
- A *protection database* that contains a list of valid user and group names, id numbers, and characteristics. Your AFS user name and ID number are the same as your UNIX user name and UID number.

Servers also ensure that AFS system configuration and binary files are replicated among themselves, and a designated server maintains a master clock that other servers and clients set their clocks by.

## 1.7 Authentication, Passwords, and Tokens

AFS does not use UNIX UIDs for authentication. To access any but public AFS files you need to have an AFS token. Almost all AFS client hosts have AFS-aware versions of `login` and `xdm`<sup>3</sup>, so a token is automatically issued when you log in.<sup>4</sup> To see what tokens you have use the `tokens` command, for example:

```
% tokens

Tokens held by the Cache Manager:

User's (AFS ID 999) tokens for afs@cs.washington.edu [Expires Apr 29 17:51]
--End of list--
```

The output shows one token for ID 999 in cell ‘`cs.washington.edu`’.

If you do not have a token, you can get one by using the `klog` command, e.g.:

```
% klog
Password: password
```

Tokens do not last forever; by default they expire 25 hours after the time issued. Use `klog`, if necessary, to renew an expired token or to replace one about to expire with a fresh one.

To change your AFS password, use `kpasswd`:<sup>5</sup>

<sup>3</sup> “AFS-aware” password handling means that you only need to provide your AFS password to be logged in

<sup>4</sup> If a token is automatically issued, a *PAG* is established as well. See Section 3.4 [PAG], page 15.

<sup>5</sup> `kpasswd` may not be available on some platforms, e.g. Linux. In that case use ‘`kas setpasswd username`’.

```
% kpasswd
Changing password for 'joeuser' in cell 'cs.washington.edu'.
Old password: old_passwd
New password (RETURN to abort): new_passwd
Retype new password: new_passwd
Password changed.
```

In order to avoid having to remember two passwords, your AFS password should be the same as your UNIX password.

## 2 File and Directory Access Controls

Each AFS *directory* (not each file) has an associated Access Control List (ACL), a list of users and groups granted or denied access to the directory's contents. Associated with each user or group on an ACL is a set of *rights* granted or denied.

### 2.1 Access Rights

AFS rights are shown in the following table. There are seven basic rights, each with a single-letter abbreviation. A set of rights is displayed as (or can be expressed as) a concatenation of the basic right abbreviations. In addition, there are some shorthand abbreviations that can be used for common combinations of rights when setting a directory's ACL:

abbrev	description
l	The <i>lookup</i> right; you need this right before you can read a directory or access any file in it.
i	The <i>insert</i> right allows you to create files or subdirectories in a directory. A newly created subdirectory inherits the the ACL of its parent.
d	The <i>delete</i> right allows you to remove files or empty subdirectories from a directory.
a	The <i>administer</i> right allows you to change the ACL of a directory. You always retain implicit administer rights to the top-level directory of any AFS volume you own (in particular your AFS home directory), so you can always restore rights to such a directory's ACL, even if you accidentally remove them.
r	The <i>read</i> right allows you to read files contained in a directory.
w	The <i>write</i> right allows you to write files contained in a directory.
k	The <i>lock</i> right allows you to run programs that need to <b>f</b> lock files in a directory.
all	All seven rights, i.e. 'rlidwka'.
read	Read and lookup rights, i.e. 'rl'.
write	All except <i>administer</i> , i.e. 'rlidwk'.
none	No rights; used to remove a user or group entry from an ACL.

## 2.2 UNIX Mode Bits

The UNIX mode bits of a directory have *no* effect in AFS, nor do the *group* and *other* mode bits of a regular file.

Only the *user* mode bits of regular files retain their function and can be used to further restrict access to a file that a directory's ACL allows access to. However,

- The UNIX owner of a file is irrelevant: the *user* bits are applied to *anyone* who can access the file.
- *Anyone* who has write access to the file via the ACL may change the mode bits with `chmod`.

*Warning: when you create a tar file from an AFS directory, be sure the mode bits are set correctly. In AFS the directory mode bits `rw-rw-rw-` mean nothing, but when untarred to a UNIX file system you will want the mode bits to be correct.*

## 2.3 AFS Protection Groups

The UNIX group of an AFS file or directory is not used. Instead, ACLs may assign (or negate) rights to AFS groups as well as individual users.

AFS provides a few predefined groups:

### `system:anyuser`

Like world permissions in UNIX: any AFS user *anywhere* can access files with the rights given this group, even without a token.

### `system:authuser`

More restrictive than '`system:anyuser`': any user who is authenticated (has a token) in the local cell can access files with the rights given this group

### `system:administrators`

Only the AFS system administrators. This group has implicit '`administer`' access to every directory.

One group local to UW CSE is also worth mentioning:

`cs-hosts` Like '`system:anyuser`', except it is restricted to processes running on CSE research hosts.

Group IDs in AFS are negative numbers, to distinguish them from UIDs.

You can create and manage your own protection groups. See man pages `pts`, `pts_creategroup`, and `pts_adduser` for more details.

## 2.4 Examining an ACL

To look at an ACL that applies to a path, use '`fs listacl path`', e.g.:

```
% cd /afs/cs/homes/joeuser/public
% fs listacl .
Access list for . is
Normal rights:
  cs-hosts rl
  joeuser rlidwka
```

This says that group `cs-hosts` has read rights and user `joeuser` has all rights.

## 2.5 Changing an ACL

To change an ACL use `'fs setacl path [id rights]...'`, e.g. to remove `'cs-hosts'` from an ACL and allow `'read'` rights to `'system:anyuser'`, enter:

```
fs setacl /afs/cs/homes/joeuser/public cs-hosts none system:anyuser read
```

It is also possible to *negate* rights on a directory, thereby denying access; see man page `fs_setacl` for details.

## 3 Additional Topics

### 3.1 Logging into a Foreign Cell

If you have an account in a foreign AFS cell, you can get a token for that account and keep it in the local host, e.g.:

```
% klog userjoe -cell transarc.com
Password: password
% tokens
Tokens held by the Cache Manager:

User's (AFS ID 5873) tokens for afs@transarc.com [Expires Apr 29 19:04]
User's (AFS ID 999) tokens for afs@cs.washington.edu [Expires Apr 29 17:51]
--End of list--
```

The `tokens` output shows tokens held for both foreign and local cells, hence files for which rights are granted may be accessed in both cells.

### 3.2 Backup Volumes and Mount Points

Each night the system takes a “snapshot” of all volumes and for each one creates a read-only backup volume that reflects conditions at the time of the snapshot. The system does not actually copy files, only pointers to files, so taking the snapshots is very efficient. After the backup volumes are generated, a backup process is run that copies them to tape, similar to the nightly backup process run on regular UNIX file systems.

A backup volume's name is that of the original with `.backup` appended, e.g. joeuser's home directory backup volume is called `user.joeuser.backup`.

To make a volume visible in the file hierarchy, an AFS *mount point* for the volume must be created via `fs mkmount directory volume`. For example, in the course of setting up joeuser's home directory, the following command was executed:

```
fs mkmount /afs/cs/homes/joeuser/oldfiles user.joeuser.backup
```

Mount points act like directories. They can be distinguished from ordinary directories with the `lsmount` opcode:

```
% cd /afs/cs/homes/joeuser
% ls -F
oldfiles/ public/
% fs lsmount *
'oldfiles' is a mount point for volume '#user.joeuser.backup'
'public' is not a mount point.
```

The initial character (in this case `#`) in a mount point's volume name determines the type of mount point, in this case a "regular mount point." See man page `fs_lsmount` for more information on mount points.

Having an `oldfiles` mountpoint available means you can retrieve accidentally deleted or hosed files yourself, *if* you discover they are gone before the next nightly backup snapshot. For example, to replace today's `public/reallyimportant` file with yesterday's:

```
cd /afs/cs/homes/joeuser
cp oldfiles/public/reallyimportant public/reallyimportant
```

### 3.3 Home Directory Structure

When your AFS home directory is first created its contents and characteristics are as follows:

- You have all rights; group `cs-hosts` has lookup rights. This means that only you can read files in your top level.
- A `public` directory is present to which you have all rights and group `cs-hosts` has read (`rl`) rights. Here go all of the files you need other individuals or processes to read, for example your `.forward` file. If such files must appear at the top home directory level, symbolic links can point to the `public` copies.
- An `oldfiles` directory is present. This directory is actually an AFS *mount point* for your home directory backup volume.

All of this is illustrated by the following example:

```

% cd /afs/cs/homes/joeuser
% fs listacl .
Access list for . is
Normal rights:
  cs-hosts l
  joeuser rlidwka
% ls -F
oldfiles/ public/
% fs listacl public
Access list for public is
Normal rights:
  cs-hosts rl
  joeuser rlidwka
% ls -l .forward
lrwxrwxrwx 1 joeuser grad_cs 15 Feb 27 1999 .forward@ -> public/.forward
% fs lsmount *
'oldfiles' is a mount point for volume '#user.joeuser.backup'
'public' is not a mount point.
% cd oldfiles
% ls -F
ls: The file ./oldfiles does not exist.
public/

```

### 3.4 Process Authentication Group

A PAG (Process Authentication Group) is a number that identifies you to the *Cache Manager*, an AFS process running on your local client host. One of the Cache Manager's jobs is to hold tokens for all authenticated users and to identify the issuers of commands that require AFS authentication. There are two ways the Cache Manager can identify you:

1. If you have established a PAG (more on how this is done below), that PAG is used. A PAG is stored in two of the kernel memory slots that UNIX uses to store groups associated with a user.<sup>6</sup>
2. If a PAG has not been established, the Cache Manager uses your UNIX UID.

You normally do not need explicitly to establish a PAG. For example, on most hosts `login` and `xdm` programs have been modified to accept your AFS password and set up a PAG and token for you automatically.

If you need a new PAG, use the `pagsh` command, which starts a Bourne shell with PAG established, e.g.:

---

<sup>6</sup> This means that if your version of UNIX allows you to be in a maximum of 16 groups and you wish to establish a PAG, the maximum number of UNIX groups is reduced to 14.

```

% groups
tech_cs bin uns ai X vlsi oti audit
% pagsh
$ exec csh                (or whatever your favorite shell is)
% groups
33536 32526 tech_cs bin uns ai X vlsi oti audit

```

Note that if the output of the `groups` command includes two adjacent numbers, there is a PAG. All child processes inherit their parent's PAG.

There can be only one token at a time per cell per host per PAG. As a consequence, when you modify tokens, the modification applies to *all* processes with the same PAG, or, if tokens are associated with your UID instead of a PAG, to *all* processes on the host under your UID that have no PAG. One of the advantages of running with a PAG is that a process running as root will not be able to use your tokens, even if it has assumed your UID.

The one token per cell per host per PAG property can lead to surprising results, say if your friend drops by to show you something and uses a shell in one of your 23 windows to run a `klog` command without issuing a `pagsh` command first!

When `pagsh` creates a new PAG your existing token(s), if any, remain associated with the old PAG or UID, and you must issue `klog` to get a new token.

The best way to ensure a long-running background job is properly authenticated and not disturbed is to start it in a new PAG with a fresh token, e.g. in a wrapper script:

```

% cd /afs/cs/homes/joeuser/public
% cat longjob.afs
#!/usr/afsws/bin/pagsh
#
# Get a fresh token
/usr/afsws/bin/klog joeuser
#
# Execute the real command
exec /afs/cs/homes/joeuser/public/longjob "$@"

```

Send mail to 'afs@cs' if you need to run background jobs exceeding 25 hours.

### 3.5 File Caching

Another function of the *Cache Manager* is to request files from servers and store them temporarily in the disk cache, a reserved area of your local host's disk. You work on the copy, which is written back to the server when the file is closed.

The size of the cache varies according to the disk capacity of individual hosts, but it generally is on the order of 50MB. If the system finds it needs more cache than the amount provided, performance will suffer. To find out how much cache is in use, enter '`fs getcacheparms`', e.g.:

```
% fs getcacheparms
AFS using 23547 of the cache's available 50000 1K byte blocks.
```

If the amount of cache in use is close to the maximum *and* you are perceiving delays in reading and writing files, send mail to ‘[afs@cs](mailto:afs@cs)’. The amount of disk available for the AFS cache varies from workstation to workstation, so increasing its size may not always be possible.

### 3.6 Privacy Flags for Groups

Each AFS user or group has a set of five *privacy flags* to indicate who has permission to view information about it and, for groups, who can make changes to the group. Only group privacy flags are discussed here.

The privacy flags are listed by ‘`pts examine group`’, e.g.:

```
% pts examine joeuser:cool
Name: joeuser:cool, id: -226, owner: joeuser, creator: joeuser,
membership: 7, flags: S-M--, group quota: 0.
```

The flags are “positional” in nature, and values for all 5 are always listed, e.g. ‘S-M--’ in the above example.

The following table shows the possible single-character values for all five group privacy flags:

1. Status flag:

- |   |  |
|---|--|
| s | Only owner and group members can view group status with ‘ <code>pts examine</code> ’ |
| S | Everyone can view group status   |

2. Owned flag:

- |   |   |
|---|---|
| - | Only owner can view the groups a group owns with ‘ <code>pts listowned</code> ’. (In AFS a <i>group</i> may own a group—including itself. See man page <code>pts_creategroup</code> for details.) |
| 0 | Everyone can view groups owned. (There is no ‘o’ value possible, since members of a group are considered owners of any groups owned by the group.)  |

3. Membership flag:

- |   |  |
|---|--|
| - | Only owner can view group membership with ‘ <code>pts membership</code> ’. |
| m | Only owner and group members can view membership                           |
| M | Everyone can view membership   |

4. Add flag:

- |   |  |
|---|--|
| - | Only owner can add users to the group with ‘ <code>pts adduser</code> ’. |
| a | Only owner and group members can add users.                              |
| A | Everyone can add users.  |

5. Remove flag:

- Only owner can remove users to the group with ‘`pts removeuser`’.
- r Only owner and group members can remove users.
- R Everyone can remove users.

Thus ‘`S-M--`’ means that all users can view status information and find out who belongs to the group, but only the owner can list groups owned, add users or remove users.

See man page `pts_setfields` for details.

### 3.7 System Type in Path Names

Several types of AFS client hosts are used in the department:

type	description
<code>alpha_osf32</code>	DEC Alpha running OSF/1 3.2
<code>i386_linux1</code>	Pentium running Linux 1.x.
<code>i386_linux2</code>	Pentium running Linux 2.x.
<code>rs_aix32</code>	IBM PowerPC running AIX 3.2.5
<code>sgi_53</code>	SGI host running IRIX 5.3
<code>sun4c_411</code>	Older Sparcstation (IPX or 2) running Sunos 4.1.x.
<code>sun4m_412</code>	Newer Sparcstation (5, 10, or 20) running Sunos 4.1.x.

If the string ‘`@sys`’ appears in an AFS path name, it will automatically be replaced by the AFS type of the local host. For example, one way to create ‘`/afs/cs/projects/ai/bin`’ directories for SunOS and IRIX platforms is as follows:

```
cd /afs/cs/projects/ai
mkdir .bin
mkdir .bin/sun4m_412 .bin/sgi_53
(cd .bin; ln -s sun4m_412 sun4c_411)
ln -s .bin/@sys bin
```

After SunOS and IRIX binaries have been copied into the ‘`sun4m_412`’ and ‘`sgi_53`’ directories, users need only make sure ‘`/afs/cs/projects/ai/bin`’ is in their PATH and the correct binaries will be available.

You can find out the system name of the current host with the ‘`fs sysname`’ command.

### 3.8 Coping with an NFS/AFS Translator

A non-AFS client host may NFS-mount `/afs` from an *NFS/AFS Translator* host. A translator host is both an AFS client and an NFS server. You can tell a “real” AFS client if the filesystem of `/afs` as reported by `df` is “AFS”:

```
% cd /afs
% df .
Filesystem      Total KB    free %used   iused %iused Mounted on
AFS              72000000 72000000    0%         0    0% /afs
```

For an NFS-mounted `/afs` you would not see “AFS” in the first column:

```
% cd /afs
% df .
Filesystem      Total    kbytes    kbytes    %
node            kbytes    used      free      used  Mounted on
curie:/afs      696832      0  696832    0%    /a/curie/afs
```

You can access publicly readable files under `/afs` on such a host without authentication.

To access protected files, you must do two things:

1. Ensure there is a token on the AFS client (translator) host. As shown above, the translator host is identified in the output of `df /afs`—host `curie` in the above example. (Obviously, you must have a UNIX account on the translator host.)
2. Use the `knfs` command on the translator host to tell its *Cache Manager* two things: the name of the NFS client machine you want to use and your UNIX uid. This lets the translator host associate the NFS client with your tokens.

This is illustrated in the following example:

```

june% hostname
june.cs.washington.edu
june% cd /afs/cs/homes/joeuser/private
june% ls -l                               (access denied to private directory)
.: Permission denied
june% rlogin curie
Last login: Mon May  6 11:37:41 from june
curie% tokens

Tokens held by the Cache Manager:

--End of list--
curie% klog
Password: password
curie% tokens

Tokens held by the Cache Manager:

User's (AFS ID 999) tokens for afs@cs.washington.edu [Expires May  7 13:04]
--End of list--
curie% knfs june 999                       (the knfs command identifies june
curie% logout                               as the NFS client and 999 as joeuser's
Connection closed.                          UNIX UID, identical to his AFS ID.)
june% dirs
/afs/cs/homes/joeuser/private
june% ls -l                               (access now granted to private directory)
total 0
-rw-rw-rw-  1 joeuser          0 May  6 11:36 private.file
june% rlogin curie
Last login: Mon May  6 11:42:18 from june
curie% knfs june 999 -unlog                 (negate effect of knfs with '-unlog' option;
curie% tokens                               but this does not remove the token)

Tokens held by the Cache Manager:

User's (AFS ID 999) tokens for afs@cs.washington.edu [Expires May  7 13:04]
--End of list--
curie% logout
Connection closed.
june% dirs
/afs/cs/homes/joeuser/private
june% ls -l                               (access denied once again)
.: Permission denied

```

The down-side to all of this (and it is a big downside) is that AFS commands are available only on true AFS clients. Hence it is advisable to use the `knfs` kludge only when really necessary.

### 3.9 The UNIX R-Commands

The UNIX “r-commands”—`rsh`, `rcp`, and `rlogin`—are used to execute commands on a remote host, copy files to/from a remote host, and login to a remote host without re-entering your UNIX password, provided a `.rhosts` file in your remote home directory allows it. AFS-aware versions of these commands are available which, in addition to these functions, set up a PAG and enable AFS authentication on a remote host, either by passing your existing AFS tokens or by establishing new tokens. To get around bugs in the r-commands provided by Transarc, we support the AFS-aware versions of the “secure shell” r-commands. See <http://www.cs.hut.fi/ssh> for more information on the Secure Shell Remote Login Program.

In the departmental standard `.cshrc` file, `PATH` is currently set such that the secure shell r-commands directory appears *after* the usual default location of `rsh`, `rcp`, and `rlogin` (`/usr/ucb`). The default path hides the secure shell r-commands because they are not transparently interchangeable with the UNIX versions; for example, if the remote host cannot handle the secure shell protocol, the AFS-aware secure shell versions revert to the standard r-commands and write the following to standard error: “Secure connection to host refused; reverting to insecure method. ...WARNING: Connection will not be encrypted.”

If you want to use the AFS-aware secure shell versions of these programs, edit your `.cshrc` to put `/afs/cs/local/bin` early in your `PATH` and `/afs/cs/local/man` early in your `MANPATH`.

### 3.10 Xlock

An AFS-aware version of `xlock`, known as `xlock.afs`, is available in `/afs/cs/local/bin`. `Xlock.afs` will get a new local token for the PAG (Section 3.4 [PAG], page 15) in which it was called when you unlock the screen by typing your AFS password. Thus, if you use `xlock.afs` to lock your workstation every day, you will not have to issue `klog` (Section 1.7 [Auth], page 10) explicitly to keep your token from expiring.

However, `xlock.afs` does not renew tokens on remote systems, so if you have used `xrsh` to open xterms or editors on remote systems, you must renew these tokens explicitly with `klog`.

## 4 When Things go Wrong

Here are some common things that can go wrong or that can cause confusion. As always, send mail to `afs@cs` if you think there is a problem that requires staff assistance.

### 4.1 AFS commands don't work.

Make sure your environment is set up correctly and you are logged into an AFS client host, as described in Section 1.3 [UNIX Env], page 7.

If `/afs/cs` is present and your environment is correct, but AFS commands are still not available, either

- the host has not been configured correctly, or
- the host is using the NFS/AFS Translator, as described in Section 3.8 [AFSNFS], page 19.

## 4.2 Your token expired in the middle of an editing session.

Use your editor's shell escape or your shell's job control to get a shell prompt so you can get a fresh token with `klog`, as described in Section 1.7 [Auth], page 10.

## 4.3 Your token expired in the middle of a background job.

Try to establish a new token immediately before running such a job, as described in Section 3.4 [PAG], page 15.

## 4.4 You cannot safely run a cron job that accesses private AFS files.

Send mail to 'afs@cs'.

## 4.5 You've lost a file.

First check to see if it's on your backup volume, as described in Section 3.2 [Backup Volumes], page 13. If not, the lost file may be on one of the backup tapes. As when requesting restoral of UNIX files, it's helpful if you supply an approximate date of last change.

## 4.6 Performance is bad.

There may be too little cache, as described in Section 3.5 [File Caching], page 16.

## 4.7 AFS seems to be stuck.

A file server may be in trouble. Use `'fs checkservers'` to check the status and `'fs whereis'` to find out which server holds the file you want.

## 4.8 You want real info on AFS, not a wimpy Beginner's Guide.

See [/afs/transarc.com/public/www/Product/AFS/FAQ/faq.html](http://afs/transarc.com/public/www/Product/AFS/FAQ/faq.html) . There are also hard copy manuals—*AFS User's Guide* and *AFS Command Reference Manual*—that you can look at, located in Sieg 117.

## Appendix A AFS Driver License Quiz Answers

1. You've used the `ls` command to look at the characteristics of file `‘.secret’` in your home directory:

```
% ls -l /afs/cs/homes/joeuser/.secret
-rw----- 1 joeuser joegroup 4698 May 1 11:25 .secret
```

True or false: no one except you and the system administrator can read this file.

**ANSWER:** Maybe true, maybe false: it depends on the Access Control List of your home directory, not on the UNIX protection modes. See Chapter 2 [AccessRights], page 11 and in particular Section 2.2 [UNIXmodes], page 12.

2. You are editing a file in your home directory and suddenly you cannot seem to save your changes. What's happened?

**ANSWER:** Your AFS token has probably timed out. See Section 1.7 [Auth], page 10.

3. The system says you have exceeded a space quota. How do you find out what your home directory space quota is?

**ANSWER:** See Section 1.6 [Servers], page 9.

4. You have a number of windows displayed on your workstation. Your friend Mikey drops by and wants to show you one of his files, which is only readable by him. He uses one of your command windows to get an AFS “token,” as follows:

```
% klog mikey
Password: mikey's_AFS_password
% view /afs/cs/homes/mikey/coolfile
```

You notice after he leaves that processes that had been working in your other windows are getting messages like

```
The file access permissions do not allow the specified action.
```

How can it be that you are denied access to your own files?

**ANSWER:** Your friend has replaced your token with his, not only in one window but very likely in all of them. See Section 3.4 [PAG], page 15.

# Index

.

.cshrc ..... 5, 7, 21  
 .rhosts ..... 20

@

@sys (in a path name) ..... 18

## A

Access Control List ..... 11, 12, 13  
 Access Right ..... 11  
 Account ..... 7  
 ACL ..... 11  
 ACL (Changing) ..... 13  
 ACL (Examining) ..... 12  
 Additional Info. .... 22  
 adduser ..... 12, 17  
 Administer (a) right ..... 11  
 All (rlidwka) rights ..... 11  
 Authentication ..... 10  
 Authentication (PAG) ..... 15  
 Availability of AFS ..... 7

## B

Background job ..... 22  
 Backup Volume ..... 13, 14, 22  
 Bad performance ..... 22

## C

Caching ..... 16  
 Cell ..... 7  
 Cell (Foreign) ..... 13  
 Changing an ACL ..... 13  
 checkservers ..... 22  
 Commands don't work ..... 21  
 Coping with an NFS/AFS Translator ..... 18  
 creatgroup ..... 12  
 Cron jobs ..... 22  
 cs-hosts group ..... 12

## D

Delete (d) right ..... 11

## E

examine ..... 17  
 Examining an ACL ..... 12  
 Expired Token ..... 22

## F

FAQ ..... 22  
 File Caching ..... 16  
 File Name ..... 7  
 Foreign Cell ..... 13  
 fs ..... 9, 12, 13, 14, 16, 22

## G

getcacheparms ..... 16  
 Getting an AFS Account ..... 7  
 Group ..... 12  
 Group (PAG) ..... 15  
 Group (Privacy Flags) ..... 17  
 groups ..... 15

## H

Help ..... 8  
 Home Directory Structure ..... 14

## I

Insert (i) right ..... 11

## K

k (Lock) right ..... 11  
 klog ..... 10, 13, 15  
 knfs ..... 18  
 kpasswd ..... 10

## L

listacl ..... 12, 14  
 listowned ..... 17  
 listquota ..... 9  
 Lock (k) right ..... 11  
 Logging into a Foreign Cell ..... 13  
 login ..... 10, 15  
 Lookup (l) right ..... 11  
 Lost files ..... 22  
 lsmount ..... 13, 14

## M

Man Page ..... 8  
 membership ..... 17  
 Mode Bits (UNIX) ..... 11  
 Mount Point ..... 13

**N**

NFS/AFS Translator .....	18
none (No rights) .....	11

**O**

oldfiles .....	14
On-Line Help .....	8

**P**

PAG .....	15
pagsh .....	15
partition .....	9
Password .....	10
Path Name (embedding system type in) .....	18
Performance is bad .....	22
Privacy Flags for Groups .....	17
Process Authentication Group .....	15
Protection Group .....	12
pts .....	12, 17
public .....	14

**Q**

quota .....	9
-------------	---

**R**

R-Commands .....	20
Read (r) right .....	11
read (rl) rights .....	11
removeuser .....	17
Response (lack of) .....	22
Right .....	11

**S**

Server .....	9
setacl .....	13
setfields .....	17
Structure of Home Directories .....	14
sysname .....	18
System Type in Path Name .....	18
system:administrators group .....	12
system:anyuser group .....	12
system:authuser group .....	12

**T**

Token .....	10
Token expires .....	22
Translator (NFS/AFS) .....	18

**U**

UNIX Environment .....	7
UNIX Mode Bits .....	11
UNIX R-Commands .....	20

**V**

Volume .....	9
Volume (backup) .....	13

**W**

write (rlidwk) rights .....	11
Write (w) right .....	11

**X**

xdm .....	10, 15
-----------	--------

# Table of Contents

<b>READ READ READ... DEMISE OF AFS FILE SERVICE .....</b>	<b>1</b>
Latest News .....	1
Executive Summary .....	1
Details .....	2
Home directories .....	2
Copying files to new NFS servers .....	2
Handling AFS mount points .....	4
<b>For the Information-Overloaded.....</b>	<b>5</b>
<b>AFS Driver License Quiz .....</b>	<b>6</b>
<b>1 Introduction .....</b>	<b>7</b>
1.1 Getting an AFS Account and Home Directory .....	7
1.2 Getting an AFS Login Directory .....	7
1.3 UNIX Environment and AFS Availability .....	7
1.4 Cells and File Names .....	8
1.5 AFS On-Line Help and Man Pages .....	8
1.6 Servers and Volumes .....	9
1.7 Authentication, Passwords, and Tokens .....	10
<b>2 File and Directory Access Controls .....</b>	<b>11</b>
2.1 Access Rights .....	11
2.2 UNIX Mode Bits .....	12
2.3 AFS Protection Groups .....	12
2.4 Examining an ACL .....	12
2.5 Changing an ACL .....	13
<b>3 Additional Topics .....</b>	<b>13</b>
3.1 Logging into a Foreign Cell .....	13
3.2 Backup Volumes and Mount Points .....	13
3.3 Home Directory Structure .....	14
3.4 Process Authentication Group .....	15
3.5 File Caching .....	16
3.6 Privacy Flags for Groups .....	17
3.7 System Type in Path Names .....	18
3.8 Coping with an NFS/AFS Translator .....	19
3.9 The UNIX R-Commands .....	21
3.10 Xlock .....	21

<b>4</b>	<b>When Things go Wrong .....</b>	<b>21</b>
4.1	AFS commands don't work.....	21
4.2	Your token expired in the middle of an editing session.....	22
4.3	Your token expired in the middle of a background job.....	22
4.4	You cannot safely run a cron job that accesses private AFS files.....	22
4.5	You've lost a file.....	22
4.6	Performance is bad.....	22
4.7	AFS seems to be stuck.....	22
4.8	You want real info on AFS, not a wimpy Beginner's Guide.....	22
	<b>Appendix A AFS Driver License Quiz Answers</b> .....	<b>23</b>
	<b>Index .....</b>	<b>24</b>